Automatic Clipping: Differentially Private Deep Learning Made Easier and Stronger

Zhiqi Bu  
AWS AI  
zhiqibu@amazon.com

Yu-Xiang Wang  
AWS AI, UC Santa Barbara  
yuxiangw@cs.ucsb.edu

Sheng Zha  
AWS AI  
zhasheng@amazon.com

George Karypis  
AWS AI  
gkarypis@amazon.com

Abstract

Per-example gradient clipping is a key algorithmic step that enables practical differential private (DP) training for deep learning models. The choice of clipping threshold $R$, however, is shown to be vital for achieving high accuracy under DP. We propose an easy-to-use replacement, called automatic clipping, that eliminates the need to tune $R$ for any DP optimizers, including DP-SGD, DP-Adam, DP-LAMB and many others. The automatic variants are as private and computationally efficient as existing DP optimizers, but require no DP-specific hyperparameters and thus make DP training as amenable as the standard non-private training. We give a rigorous convergence analysis of automatic DP-SGD in the non-convex setting, which shows that it can enjoy an asymptotic convergence rate that matches the standard SGD, under a symmetric gradient noise assumption of the per-sample gradients. We also demonstrate on various language and vision tasks that automatic clipping outperforms or matches the state-of-the-art, and can be easily employed with minimal changes to existing codebases.

1 Introduction

Deep learning has achieved impressive progress in a wide range of tasks. These successes are made available, in part, by the collection of large datasets, sometimes containing sensitive private information of individual data points (e.g., chest scan images, DNA sequences). Prior works have illustrated that deep learning models pose severe privacy risks to individual subjects in the training data and are susceptible to various practical attacks. For example, machine learning services such as Google Prediction API and Amazon Machine Learning can leak membership information from the purchase records [62]; if one feeds the GPT2 language model with some specific prefix, the model will autocomplete texts that contain someone’s full name, phone number, email address, etc., from the training data that it memorizes [11].

Differential privacy (DP) [23, 25, 24] is a formal definition of privacy that has been shown to prevent the aforementioned privacy risks in deep learning [1]. On a high level, the key difference between the DP deep learning and the regular one is whether the gradient is privately released. In other words, while the standard optimizers update on the summed gradient $\sum_i g_i$, and DP optimizers update on the private gradient:

$$\text{DP Optimizer}(\{g_i\}_{i=1}^B) = \text{Optimizer}(\sum_i g_i \cdot \text{Clip}(\|g_i\|; R) + \sigma R \cdot N(0, I))$$  (1.1)

$$\text{Standard Optimizer}(\{g_i\}_{i=1}^B) = \text{Optimizer}(\sum_i g_i)$$  (1.2)

Here $g_i \in \mathbb{R}^d$ is the per-sample gradient of loss $l_i$, $N$ is the standard normal, $\sigma$ is the noise multiplier, and $R$ is the clipping threshold. The clipping function $\text{Clip} : \mathbb{R}^d \to \mathbb{R}$ is defined such that
\( \|g_i \cdot \text{Clip}(g_i; R)\| \leq R \). For instance, the DP-SGD in [1] on batch \( B_i \) is

\[
\text{DP-SGD}_{\text{Abadi}} : \quad w_{t+1} = w_t - \eta \left( \sum_{i \in B_t} \frac{\partial l_i}{\partial w} \min \left( \frac{R}{\| \frac{\partial l_i}{\partial w} \|}, 1 \right) + \sigma R \cdot \mathcal{N}(0, I) \right)
\]

(1.3)

In comparison to the regular training (1.2), two additional DP-specific hyperparameters \( R \) and \( \sigma \) need to be determined in DP learning (1.1). On the one hand, setting the noise multiplier \( \sigma \) is easy and can be derived analytically prior to the training. Whenever the privacy budget \((\epsilon, \delta)\) is determined, one can apply off-the-shelf privacy accounting tools in Section 2.1 to determine \( \sigma \), based on the subsampling probability \( p \) and the number of iterations \( T \):

\[
\text{privacy} \_\text{accountant}(\sigma, p, T; \delta) = \epsilon
\]

On the other hand, the choice of clipping threshold \( R \) is crucial to the performance of DP models, yet the hyperparameter tuning is much labor-intensive. Recent advances of DP deep learning on ImageNet [37] and on E2E datasets [40], using ResNet18 and GPT2 respectively, illustrate that the performance is very sensitive to \( R \). We have reproduced their results in Figure 1. Observe that on ImageNet, ResNet18 can drop from the highest 45% accuracy to 31% if \( R \) is chosen 2 times larger, and to 0.1% if \( R \) is chosen 4 times larger. Similar drastic drop can also be observed in [37, Figure 3] even if the noise multiplier \( \sigma = 0 \). Unlike the noise multiplier \( \sigma \), the clipping threshold \( R \) cannot be inferred from the privacy budget \((\epsilon, \delta)\) and have to be tuned. Consequently, DP training necessarily requires an expensive 2D grid search for \((R, \eta)\), like Figure 1, whereas the regular training only requires an easy 1D grid search for \( \eta \). Even worse, the difficulty of tuning a per-layer clipping threshold vector [47], i.e. one clipping threshold for one layer, may increase exponentially as the number of layers increases.

To save the effort of tuning \( R \), previous researches have proposed different approaches. In [3, 55, 27, 30], researchers advocate to use data-adaptive information to select \( R \), such as a specified quantile of the gradient norm distribution. These adaptive clipping methods can be a little ad-hoc: they often replace the need to tune \( R \) by the need to tune one or more new hyperparameters, e.g. the quantile to use and the ratio to split the privacy budget between the quantile decision and the gradient perturbation. Another approach used by the practitioners is to replace the single 2D grid search by multiple cheaper 1D grid searches. For example, the researchers propose, in [37, Section 3.3] to fine-tune \( \eta \) with non-DP SGD, fix \( \eta \) and sweep over various values of the clipping threshold \( R \) with DP-SGD, then further fix \( R \) and do one more grid search on \( \eta \). However, tuning \( R \) formally in a data-dependent way (e.g. through cross-validation) introduces additional privacy loss [52], and most existing empirical work does not privately conduct hyperparameter tuning.

We take a completely different route by proposing a new clipping principle that removes \( R \), instead of coming up with methods to find the appropriate \( R \). We term our method as automatic clipping and we term the versions of DP optimizers using it as automatic DP optimizers. Our contributions are:

1. We propose the automatic clipping in (4.1) that expunges the clipping threshold from general DP optimizers, making DP training as amenable as regular training. In large scale (DP GPT-level) like Figure 1, automatic clipping can reduce the computation of ablation study by \( 5 \times 1 \).
2. We show that automatic DP optimizers are as private and efficient as existing DP optimizers.
3. We show in Theorem 4 that automatic DP-SGD converges in the non-convex setting, at the same asymptotic convergence rate as the standard SGD. Our theoretical analysis successfully explains the training behaviors of deep learning in previous empirical works.
4. We demonstrate the superiority of automatic clipping on a variety of vision and language tasks, especially with large models including ResNet, RoBERTa and GPT2.
5. In Appendix K, we include simple code snippets that demonstrate how easy it is to switch from Abadi’s clipping to our automatic clipping in popular codebases, e.g. Opacus and ObJAX.

## 2 Preliminaries

### 2.1 Differential Privacy

We consider the \((\epsilon, \delta)\)-DP in Definition 2.1, where smaller \((\epsilon, \delta)\) means stronger privacy guarantee.

\footnote{The hyperparameter tuning of \((R, \eta)\) takes days (e.g. GPT2 [40]) to months (e.g. GPT3-175B) on large foundation models, highlighting the significance of our method to expunge the additional \( R \).}
Learning rate

\[
\text{Definition 2.1} \quad \text{A randomized algorithm } M \text{ is } (\varepsilon, \delta)\text{-differentially private (DP) if for any two neighboring}\footnote{\text{S}^\prime \text{ is a neighbor of } S \text{ if one can obtain } S^\prime \text{ by adding or removing one data point from } S.} \text{ datasets } S, S', \text{ and for any event } E, \]

\[
\mathbb{P}[M(S) \in E] \leq e^\varepsilon \mathbb{P}[M(S') \in E] + \delta. \tag{2.1}
\]

In words, DP restricts the influence of an arbitrary sample, so that the information contributed by such sample is limited and less vulnerable to privacy attacks. In deep learning, DP is achieved by applying the \textit{subsampled Gaussian mechanism} to privatize the minibatch gradients during training.

As illustrated in Equation (1.1), the subsampled Gaussian mechanism involves (1) Sampling a minibatch by including each data point iid with probability \( p \) (2) per-sample gradient clipping to bound the \( l_2 \) norm sensitivity at \( R \) and (3) adding independent Gaussian noise proportional to the sensitivity \( R \) and \( \sigma \), which is derived from the privacy loss \( \epsilon \). This can be realized by leveraging a variety of modern privacy accounting tools, such as Renyi DP (or moments accountant) \[1, 49, 68\], Privacy Loss distribution (Fourier accountants) \[36, 29, 76\], or Gaussian DP \[18, 7\].

2.2 Differentially Private optimizers with general clipping operations

Privately released stochastic gradients (through the Gaussian mechanism) can be used to instantiate various off-the-shelf optimizers, which gives rise to DP-SGD in (1.3), DP-HeavyBall, DP-AdaGrad, DP-Adam, DP-FedAvg, DP-FedSGD \[47\], etc. To improve the performance of DP optimizers, previous researches can be classified into two categories.

The first category, where the majority of researches lie in, works with Abadi’s clipping and focuses on better design of \( R \). To name a few examples, one can adaptively design \( R_t \) for each iteration \( t \) \[3, 55, 27\], or design the per-layer clipping threshold vector \( \mathbf{R} \in \mathbb{R}^L \) for \( L \) layers \[1, 47\] so as to apply a different clipping threshold for each layer.

Fewer works fall into the second category that proposes new clipping methods. In fact, any function \( \text{Clip}_p : \mathbb{R}^d \to \mathbb{R} \) satisfying \( \| \text{Clip}_p(g) \cdot g \| \leq R \) can serve as a valid clipping function besides Abadi’s. For instance, the global clipping \[9\] proposes \( \text{Clip}_p^\text{global}(g) := I(\|g\| < R) \) to mitigate the bias of the private gradient and alleviate the mis-calibration issue of DP classifiers. Our automatic clipping also belongs to this category. We note that different clipping methods work orthogonally to optimizers, network architectures and gradient norm computation (see Section 7).

3 Motivation

3.1 Small clipping threshold often works best

One intriguing observation that we can make about the recent studies on DP learning with large models is that the state-of-the-art (SOTA) results are often achieved with very small clipping threshold \( R \). This observation is consistent in both vision and language tasks. In \[40\], GPT2 (about 800 million parameters) and RoBERTa models (over 300 millions parameters) achieve the best results under DP on QNLI, MNLI, SST-2, QQP, E2E, and DART datasets, with each per-sample gradient clipped to length \( R = 0.1 \).

In \[37, 16, 48\], ResNets and Vision Transformers achieve the best DP results on ImageNet with \( R = 1 \); in \[65\], the best DP results on CIFAR10 use \( R = 0.1 \) with ResNeXt-29 and SimCLRv2 \[13\]. The effectiveness of small clipping threshold together with proper learning rate is depicted in Figure 1.

Intuitively, smaller \( R \) implies that the Abadi’s clipping (3.1) happens, which means \( \min \left( R/\|g_i\|, 1 \right) = R/\|g_i\| \). Given that the clipping threshold \( R \) is so small compared to the number of parameters in large neural networks, and that strong DP is guaranteed when the number of training iterations is small (i.e. \( \|g_i\| \) has not converged to small values yet), we expect and empirically observe that the clipping happens on a large proportion of per-sample gradients at all iterations. For instance, we find in the GPT2 generation experiments in \[40\] that 100% of per-sample gradients are clipped...
at all iterations; in classification tasks such as QQP/QNLI/MNLI, the percentage of clipping is about 20 ~ 60% on average (more details in Appendix H.1).

3.2 Per-sample gradient normalization as new clipping

In the small clipping threshold regime, we can approximately view

\[ \text{Clip}_{\text{Abadi}}(g_i; R) = \min \left( \frac{R}{\|g_i\|}, 1 \right) \approx \frac{R}{\|g_i\|} =: \text{Clip}_{\text{AUTO-V}}(g_i; R) \]  

(3.1)

and thus derive a novel private gradient \( \sum_{i \in B_t} R \frac{g_i}{\|g_i\|} + \sigma R \cdot N(0, I) \). Here AUTO-V stands for the vanilla automatic clipping, which essentially performs the normalization on each per-sample gradient. As a specific example, we can write the \( R \)-dependent automatic DP-SGD as

\[ R\text{-dependent DP-SGD}_{\text{AUTO-V}}: w_{t+1} = w_t - \eta \left( \sum_{i \in B_t} R \frac{\partial l_i}{\partial w_i} / \| \partial l_i / \partial w_i \| + \sigma R \cdot N(0, I) \right) \]  

(3.2)

We may view our AUTO-V clipping as to maximize the dot-product similarity (a commonly used similarity measure, e.g. in the attention block in transformers [66]) between the clipped gradient and the regular gradient. Suppose we want

\[ \max_{C_t} \left\{ \sum_i C_i g_i, \sum_j g_j \right\} \text{ s.t. } 0 \leq C_i \leq R/\|g_i\| \]

Note that the constraint is a sufficient condition for clipping, as discussed in Section 2.2. It is not hard to see that the optimal clipping factor is

\[ C_t = R/\|g_i\| \cdot I(\langle g_i, \sum_j g_j \rangle > 0) \]

If the per-sample gradients are indeed concentrated in the sense \( \forall i, \langle g_i, \sum_j g_j \rangle \geq 0 \), then AUTO-V is the optimal per-sample gradient clipping. We compare with Abadi’s clipping in Figure 2, where the dot-product similarity is significantly magnified by our AUTO-V clipping.

Figure 2: RoBERTa-base with DP-Adam (\( \epsilon = 3 \)) on SST2 dataset, as in Section 6.2.

3.3 Stability constant breaks scale-invariance and remains stationary

One potential drawback of AUTO-V clipping is that all gradients lose their magnitudes information completely, since \( \|g_i \cdot \text{Clip}_{\text{AUTO-V}}(g_i; R)\| = R, \forall i \). This scale-invariance in AUTO-V and partially in Abadi’s clipping (when \( \|g_i\| > R \)) leads to the "lazy region" issue: the parameters will not be updated by DP-GD even if the true gradients are non-zero. In Figure 3, we illustrate in a logistic regression\(^3\) that AUTO-V and Abadi’s clipping have zero clipped gradient when the trainable parameter \( \theta \in [-2, 2] \), as the gradients from two classes cancel each other.

To preserve the magnitude information and thus escape the lazy region, we propose the AUTO-S clipping, with a positive stability constant \( \gamma \):

\[ \text{Clip}_{\text{AUTO-S}}(g_i; R) := R/(\|g_i\| + \gamma) \]  

(3.3)

We visualize in Figure 4 that AUTO-S allows larger per-sample gradients to have larger magnitudes after the clipping, while still allowing smaller gradients to vanish after “clipping”. That is, as \( g_i \to 0 \), the existence of \( \gamma \) allows the clipped gradient \( C_t g_i \to g_i/\gamma \) rather than having a magnitude \( R \) as in AUTO-V. We elaborate this point in Section 4.3. This is critical in our convergence analysis and allows DP-SGD\(_{\text{AUTO-S}}\) (but not DP-SGD\(_{\text{AUTO-V}}\)) to converge to zero gradient norms in Section 5.

\(^3\)The settings are in Appendix F, where the lazy region issues also emerge in the mean estimation problem. We note that the lazy region is also discussed in [14, Example 2].
4 Automatic DP Training

One may wonder why our clipping (3.1)(3.3) is automatic at all, if the hyperparameter \( R \) is still present and there is an additional parameter \( \gamma \) to choose. It turns out that any constant choice of \( R > 0 \) is equivalent to choosing \( R = 1 \), and common deep learning optimizers are insensitive to the choice of \( \gamma \) (e.g. for any \( \gamma > 0 \), we show that the gradient norm converges to zero at the same asymptotic rate in Theorem 4; see also the ablation study in Figure 14). Consequently, we set \( \gamma = 0.01 \) as the default. Specifically, let us redefine the \( R \)-independent clipping function:

\[
C_{\text{11PAUTO-S}}(g_i) := 1/(\|g_i\| + \gamma).
\] (4.1)

With this clipping, we can design automatic DP optimizers similar to (1.1):

\[
\text{Automatic DP Optimizer}(\{g_i\}_{i=1}^{B}) = \text{Optimizer} \left( \sum_{i \in B_I} \frac{g_{i,t}}{\|g_{i,t}\|} + \sigma \cdot \mathcal{N}(0,I) \right)
\] (4.2)

Clearly, the new private gradient \( \hat{g}_t \) from our automatic clipping is \( R \)-independent, in contrast to the one used in (1.1). A concrete example (in the case of \( \gamma = 0 \)) that is comparable to (3.2) will be

\[
R\text{-independent DP-SGD}_{\text{AUTO-V}} : \quad w_{t+1} = w_t - \eta \left( \sum_{i \in B_I} \frac{\partial l_i}{\partial w_t} / \|\hat{g}_{i,t}\| \right) + \sigma \cdot \mathcal{N}(0,I)
\] (4.3)

Leveraging the private gradient \( \hat{g}_i \) in (4.2), we can train DP neural networks without tuning DP-specific hyperparameters \( R \) and \( \sigma \), as demonstrated in Algorithm 1.

Algorithm 1 Automatic Deep Learning with DP

**Parameters:** initial weights \( w_0 \), learning rate \( \eta_t \), sampling probability \( p \), number of iterations \( T \).

1. Find \( \sigma \) such that \( \epsilon_{\text{Accountant}}(\delta, \sigma, p, T) \leq \epsilon \) from any privacy accountant.
2. for iteration \( t = 1, \ldots, T \) do
3. Sample a batch \( B_t \) by including each data point i.i.d. with probability \( p \).
4. Apply automatic clipping to per-sample gradients \( \{g_i\}_{i \in B_I}; \hat{g}_i = g_i/(\|g_i\|_2 + 0.01) \).
5. Add Gaussian noise to the sum of clipped gradients: \( \hat{g} = \sum_i \hat{g}_i + \sigma \cdot \mathcal{N}(0,I) \).
6. Update \( w_t \) by any optimizer on the private gradient \( \hat{g} \) with learning rate \( \eta_t \).

We will elaborate two distinct reasons in the next sub-sections for the following statement:

\[
\text{DP Optimizer}_{\text{Abadi}} \approx \text{R-dependent DP Optimizer}_{\text{AUTO}} \equiv \text{R-independent DP Optimizer}_{\text{AUTO}}
\]

which reduces the hyperparameter tuning of DP training to that of the regular training, i.e. only on learning rate, weight decay, etc. The significant save in the tuning effort is illustrated in Figure 15.

4.1 Non-adaptive optimizer couples clipping threshold with learning rate

With \( R \)-dependent automatic clipping, DP-SGD becomes

\[
w_{t+1} = w_t - \eta \left( \sum_{i \in B_I} \frac{R}{\|g_{i,t}\| + \gamma} + \sigma R \cdot \mathcal{N}(0,I) \right) = w_t - \eta R \hat{g}_t.
\]

We can view \( \eta_{\text{effective}} \equiv \eta R \) as a whole: increasing \( R \) has the same effect as increasing \( \eta \), which explains the diagonal pattern in Figure 1(lower plot) where DP-SGD_{\text{Abadi}} is applied with small clipping threshold\(^4\). We extend to general non-adaptive optimizers in Theorem 1\(^5\).

**Theorem 1.** Non-adaptive \( R \)-dependent automatic DP optimizers (including SGD, Heavyball[56] and NAG[51]), with learning rate \( \eta \) and weight decay \( \lambda \), is equivalent to \( R \)-independent automatic DP optimizers, with learning rate \( \eta R \) and weight decay \( \lambda R \).

4.2 Adaptive optimizer can be insensitive to clipping threshold

Adaptive automatic DP optimizers are different than the non-adaptive ones, as the clipping threshold cancels out instead of being coupled with learning rate. To see this, we scrutinize DP-Adam_{\text{Abadi}} (which is similar to DP-Adam_{\text{AUTO-V}}) in Figure 1(upper plot), where columns to the left are almost

\(^4\)When we further consider weight decay in automatic clipping (included in Theorem 1), increasing \( R \) is no longer equivalent to increasing \( \eta \), as \( \eta \) also couples with the weight decay constant \( \lambda \).

\(^5\)This coupling of \( \eta \) and \( R \) is also partially observed in [16] through a reparameterization trick of Abadi’s clipping. Unlike AUTO-S/V, their coupling is not strict (e.g. doubling \( R \) is not equivalent to doubling \( \eta \) in their Figure 8, thus necessitating tuning both \( (\eta, R) \)), and the relationship to weight decay was not discussed.
identical. Further evidence is observed in [48, Table 5] that shrinking $R$ has zero effect on LAMB. We now give a simple explanation using AdaGrad [21]:

$$w_{t+1} = w_t - \eta \frac{g_t}{\sqrt{\sum_{\tau<t} g^2_\tau}}$$

where $g_t = \sum_i g_{t,i}$ is the gradient sum. In $R$-dependent DP-AdaGrad$_{AUTO-V}$, the private gradient is $R\hat{g}_i$ in place of the standard gradient sum $g_i$:

$$w_{t+1} = w_t - \eta \frac{R\hat{g}_i}{\sqrt{R^2 \sum_{\tau<t} (\hat{g}_\tau)^2}} = w_t - \eta \frac{\hat{g}_i}{\sqrt{\sum_{\tau<t} (\hat{g}_\tau)^2}}.$$  

We generalize to the general adaptive optimizers in Theorem 2.

Theorem 2. Adaptive $R$-dependent automatic DP optimizers (e.g. AdaGrad[21], AdaDelta[75], AdaMax/Adam[34], NAdam[19], RAdam[42], LARS[71], LAMB[72]), with learning rate $\eta$ and weight decay $\lambda$ is equivalent to $R$-independent automatic DP optimizers with learning rate $\eta$ and weight decay $\lambda/R$. With decoupled weight decay[45], $R$-dependent automatic DP-AdamW is equivalent to $R$-independent automatic DP-AdamW with the same $\eta$ and $\lambda$.

In Appendix B.3, we also analyze the automatic DP optimizers with per-layer clipping style.

4.3 Automatic clipping is equally private and maximizes utility

In Theorem 3 (proved in Appendix A), we show that the new private gradient $\hat{g}_i$ in (4.2) has the same level of privacy guarantee as the existing one in (1.1), since the global sensitivity remains the same (see Figure 4). We note that as long as $\gamma > 0$, the magnitude information of per-sample gradients is preserved by AUTO-S, in the sense that $\|g_i\| > \|\hat{g}_i\| \iff \|C_i g_i\| > \|C_i \hat{g}_i\|$, whereas this can be violated in both the AUTO-V and Abadi’s clipping (as depicted by the flat curve in Figure 4 when $\|g_i\| > 1$). Additionally, note that when $\gamma$ is small, almost all data points “max out” the signal relative to the amount of noise we add. To say it differently, for the same amount of noise, AUTO-S with small $\gamma$ allows more signal to be pushed through a differentially private channel. Towards the end of the training, i.e., at the limit when $\|g_i\| \to 0$ for all $i$, then we have $\sum_i \frac{g_i}{\|\hat{g}_i\| + \gamma} \to \frac{1}{\gamma} \sum_i g_i$. In words, the clipped gradients become closer to the standard SGD, thus do not suffer from the instability of AUTO-V.

Theorem 3. Under the noise multiplier $\sigma$, number of iterations $T$, subsampling probability $B/n$, DP optimizers using AUTO-V or AUTO-S clipping satisfy $(\epsilon_{Accountant}(\delta, \sigma, B/n, T), \delta)$-DP, where $\epsilon_{Accountant}$ is any valid privacy accountant for DP-SGD under Abadi’s clipping.

4.4 Comparison to related works on clipping

A concurrent work [15] also proposes to normalize gradients (same as AUTO-V) in the distributed setting, instead of clipping them. Similarly, [16] re-parameterizes the per-sample gradient clipping to mimic the normalization. However, $R$ remains and is expensive to tune (see [16, Figure 8]). In sharp contrast, we (1) completely remove $R$ to save the tuning efforts in Figure 15, (2) introduces the stability constant (i.e. AUTO-S) with theoretical (see Section 3.3) and empirical benefits (see Section 6), (3) formalizes the understanding of automatic DP optimizers, e.g. in Footnote 4&5, and (4) provides the non-convex convergence analysis and NLP experiments absent from [15, 16].

5 Convergence analysis of DP-SGD with automatic clipping

5.1 Convergence theory of DP-SGD to stationary points

We highlight that automatic clipping can be more amenable to analysis than Abadi’s clipping in [14], since we no longer need to decide whether each per-sample gradient is clipped.

To analyze the convergence of automatic DP-SGD (4.2) in the non-convex setting, we follow the standard assumptions in the SGD literature [26, 2, 6], including a symmetry assumption on the gradient noise, which is empirically verified in [14, Figure 3]6.

6This symmetry assumption is relaxed from the Gaussian noise assumption (since a zero-mean Gaussian is symmetric) in the SGD literature [46, 63, 12, 70]. By setting minibatch size to 1, we reduce the noise assumption to a per-sample gradient case.
Assumption 5.1 (Lower bound of loss). For all $w$ and some constant $L_*$, we have $L(w) \geq L_*$.

Assumption 5.2 (Smoothness). Let $\nabla L(w)$ denote the gradient of the objective $L(w)$. Then $\forall w, v$, there is an non-negative constant $L$ such that
\[ L(v) - [L(w) + \nabla L(w) \cdot (v - w)] \leq \frac{L}{2} \| v - w \|^2. \quad (5.1) \]

Assumption 5.3 (Gradient noise). The per-sample gradient noise $\tilde{g}_{t,i} - g_i$ is i.i.d. from some distribution such that
\[ \mathbb{E}(\tilde{g}_{t,i} - g_i) = 0, \quad \mathbb{E}\| \tilde{g}_{t,i} - g_i \|^2 \leq \xi^2, \]
and $\tilde{g}_{t,i}$ is centrally symmetric about $g_i$ in distribution: $\tilde{g}_{t,i} - g_i = g_i - \tilde{g}_{t,i}$.

We show in Theorem 4 that DP-SGD with AUTO-S clipping allows the true gradient norm to converge to zero, though the clipped gradient may still be biased, but not so with AUTO-V clipping. We leave the proof in Appendix C.1.

**Theorem 4.** Under Assumption 5.1, 5.2, 5.3, running DP-SGD with automatic clipping for $T$ iterations and setting the learning rate $\eta \propto 1/\sqrt{T}$ give\(^3\)
\[
\min_{0 \leq t \leq T} \mathbb{E}(\|g_t\|) \leq \mathcal{G} \left( \frac{4}{\sqrt{T}} \left( L_0 - L_* \right) L \left( 1 + \frac{\sigma^2 d}{B^2} \right) ; \xi, \gamma \right) := \min_{r \geq 0} \frac{\xi}{r} + \mathcal{F} (\cdots ; r, \xi, \gamma). \quad (5.2)
\]

Here $\cdots$ represents the first argument of $\mathcal{F}$, and $\mathcal{G}$ is increasing and positive. As $T \to \infty$, we have $\min_t \mathbb{E}(\|g_t\|) = O(T^{-1/2})$ for AUTO-S, the same rate as the standard SGD given in Theorem 9.

We now analyze the many factors that affect the convergence in Theorem 4, from a unified viewpoint of both the convergence and the privacy.

We start with the stability constant $\gamma$ and the learning rate $\eta_t$, both only affect the convergence not the privacy. We empirically observe in Figure 7 that small $\gamma$ benefits the convergence at initial iterations (when the privacy guarantee is strong) but larger $\gamma$ converges faster asymptotically. For $\eta_t$, the optimal is in fact the minimizer of the hyperbola in (C.4), that is unique and tunable.

Next, we focus on the hyperparameters that affect both convergence and privacy: the batch size $B$, the noise multiplier $\sigma$, and the number of iterations $T$. These hyperparameters have to be considered along the privacy-accuracy tradeoff, not just from a convergence perspective.

Recall that given a fixed privacy budget $(\epsilon, \delta)$, we rely on modern privacy accountant for computing the appropriate combinations of parameter $\sigma, T, B$. The exact expression of the bound as a function of $(\epsilon, \delta)$ is somewhat messy. For this reason, we illustrate our analysis in terms of the surrogate parameter $\mu$ for $\mu$-GDP [18]. [7] showed that DP-SGD’s privacy guarantee asymptotically converges to $\mu$-GDP (as $T \to \infty$) with $\mu = \frac{B}{n} \sqrt{T(e^{1/\sigma^2} - 1)}$. $\mu$-GDP implies $(\epsilon, \delta)$-DP with $\epsilon = \mu^2 + \mu \sqrt{2 \log(1/\delta)}$\(^4\).

We can alternatively leverage $\rho$-tCDP [10] for similar conclusions, using $\rho$ in place of $\mu^2$ in (5.3).

\(^3\)The upper bound takes an implicit form of $\mathcal{G}(\cdots ; \xi, \gamma)$ because it is a lower envelope of functions $\frac{\xi}{r} + \mathcal{F}(\cdots ; r, \xi, \gamma)$ over all possible $r > 0$, whose forms are detailed in Theorem 6. Notice that $\mathcal{G}$ results only from the clipping operation, not from the noise addition.

\(^4\)More precisely, $\mu$-GDP is equivalent to an entire family of $(\epsilon, \delta)$-DP for any $\epsilon > 0$ and $\delta = \Phi(\mu/2 - \epsilon/\mu) - \epsilon \Phi(-\mu/2 - \epsilon/\mu)$ where $\Phi$ is the standard Gaussian CDF.
Theorem 5. Under Assumption 5.1, 5.2, 5.3, fixing the asymptotic \( \mu(\epsilon, \delta)-\)DP parameter, running DP-SGD with automatic clipping for \( T \) iterations and setting the learning rate \( \eta \propto 1/\sqrt{T} \) give

\[
\min_{0 \leq t \leq T} \mathbb{E}(\|g_t\|) \leq G \left( 4 \sqrt{(L_0 - L_*)L} \left( \frac{1}{T} + \frac{d}{\mu^2 n^2} + O\left( \frac{1}{B^2 T} \right) \right); \xi, \gamma \right)
\]

(5.3)

To show that our analysis matches the training behaviors observed in SOTA empirical work [40, 37, 16, 65, 48, 74], we minimize the first argument of \( G \) in (5.3), denoted as \( X(B, T, \mu, d, L, L_0) \).

1. [Train longer with larger noise] Fixing the expected batch size \( B \), we see that \( X \) is decreasing in \( T \). Hence larger \( T \) and consequently larger \( \sigma \) are preferred.

2. [Larger batch size helps] Fixing number of iterations \( T \) or epochs \( E = BT/n \), we see that \( X \) is decreasing in \( B \). Hence larger \( B \) and consequently larger \( \sigma \) are preferred.

3. [Pretraining is critical] Pretraining can boost the DP accuracy through a much smaller initial loss \( L_0 \) and from a smooth (small \( L \)) and flat (small \( \xi \), c.f. Figure 7(left)) initialization.

4. [Learning rate needs tuning] The optimal learning rate by minimizing (C.4) is \( \eta^* = \sqrt{\frac{(L_0 - L_*)\mu^2 n^2}{L(\mu^2 n^2 + dT)}} \).

This indicates that one should use larger learning rate for smaller model \( d \), weaker privacy (larger \( \mu \) or small \( \epsilon \)), or smaller iteration budget \( T \).

6 Experiments

We evaluate our automatic DP training on image classification, sentence classification, and table-to-text generation tasks. Detailed settings including hyperparameters can be found in Appendix G.

6.1 Image classification

For MNIST/FashionMNIST, we use the same setup as in [53, 65, 61] with a simple CNN. For CIFAR10, we use the same setup as in [65] with pretrained SimCLRv2 [13]. For ImageNet, a 10-class sub-task of ImageNet [17], we use the same setup as in [35] without the learning rate decay. For CelebA [44], the real human face dataset, we train ResNet9 [31] with group normalization to replace the batch normalization. Notice that CelebA contains high-resolution (178x218) images, each with 40 labels. We consider CelebA for either multi-class classification on one label, e.g. ‘Smiling’ and ‘Male’, or for multi-label/multi-task problem to learn all labels simultaneously.

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
<th>(( \epsilon, \delta ))</th>
<th>( \epsilon ) Accuracy %</th>
<th>( \epsilon = \infty ) Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>4-layer CNN</td>
<td>(1, 1e-5)</td>
<td>98.04 \pm 0.09 \quad 98.15 \pm 0.07</td>
<td>98.11 \pm 0.07</td>
</tr>
<tr>
<td>FashionMNIST</td>
<td>4-layer CNN</td>
<td>(3, 1e-5)</td>
<td>86.04 \pm 0.26 \quad 86.36 \pm 0.18</td>
<td>89.57 \pm 0.13</td>
</tr>
<tr>
<td>CIFAR10 pretrained SimCLRv2</td>
<td>(2, 1e-5)</td>
<td>92.44 \pm 0.13 \quad 92.70 \pm 0.02</td>
<td>94.42 \pm 0.01</td>
<td></td>
</tr>
<tr>
<td>ImageNet</td>
<td>ResNet9</td>
<td>(5, 1e-4)</td>
<td>60.29 \pm 0.34 \quad 60.71 \pm 0.48</td>
<td>71.11 \pm 0.37</td>
</tr>
<tr>
<td>CelebA [Smiling]</td>
<td>ResNet9</td>
<td>(5, 1e-6)</td>
<td>90.75 \pm 0.11 \quad 91.18 \pm 0.08</td>
<td>92.61 \pm 0.20</td>
</tr>
<tr>
<td>CelebA [Male]</td>
<td>ResNet9</td>
<td>(8, 3e-6)</td>
<td>95.54 \pm 0.14 \quad 95.70 \pm 0.07</td>
<td>97.90 \pm 0.01</td>
</tr>
<tr>
<td>CelebA Multi-label</td>
<td>ResNet9</td>
<td>(8, 3e-6)</td>
<td>86.81 \pm 0.03 \quad 87.05 \pm 0.01</td>
<td>90.30 \pm 0.02</td>
</tr>
<tr>
<td>CelebA Multi-label</td>
<td>ResNet9</td>
<td>(8, 5e-6)</td>
<td>87.52 \pm 0.15 \quad 87.58 \pm 0.04</td>
<td>90.30 \pm 0.02</td>
</tr>
</tbody>
</table>

Table 1: Average test accuracy and 95% confidence interval on image tasks over 5 runs.

In Table 1, we observe that AUTO-S clipping outperforms existing clipping in all datasets with statistical significance. Interestingly, the standard deviation from different runs is smaller for automatic DP optimizers, indicating better reproducibility and stability. We additionally experiment 40 binary classification problems on CelebA with respect to each label, and observe that the mean accuracy further improves to 91.63% at \( \epsilon = 8 \) for AUTO-S (see Appendix J).

6.2 Sentence classification

On five benchmark language datasets (MNLI(m/mm)[69], QQP[33], QNLI[59], SST2[64]), we compare our automatic DP training with reparameterized gradient perturbation (RGP, [74]) and full-parameter finetuning (full, [40]) using RoBERTa models [43]. These methods use the same experimental setup. For language models, our automatic training is based on the codebase of [40].

*See https://github.com/lxuechen/private-transformers and our modification in Appendix K.3.*
In Table 2 and Table 3, we note that full parameter finetuning with AUTO-S outperforms or at least matches SOTA on all tasks. We use exactly the same hyperparameters as in [40].

6.3 Table-to-text generation

We compare our automatic DP training with a variety of fine-tuning methods, for table-to-text generation task on E2E dataset [22], where the goal is to generate texts about different aspects of a restaurant’s data. We measure the success on this task by BLEU, ROUGE-L (in Table 4), METEOR, NIST, CIDEr (extended in Table 8), with higher value meaning better model quality.

Table 4: Test performance on E2E dataset with GPT2. Additional performance measures are included in Table 8. The best two GPT2 models for each row are marked in bold.

<table>
<thead>
<tr>
<th>Metric</th>
<th>DP guarantee</th>
<th>GPT2 large</th>
<th>GPT2 medium</th>
<th>GPT2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\epsilon = 3$</td>
<td>$\epsilon = 8$</td>
<td>$\epsilon = \infty$ (non-DP)</td>
<td>$\epsilon = 3$</td>
</tr>
<tr>
<td>BLEU</td>
<td>full AUTO-S</td>
<td>full AUTO-S</td>
<td>full AUTO-S</td>
<td>full AUTO-S</td>
</tr>
<tr>
<td></td>
<td>64.180</td>
<td>63.850</td>
<td>63.850</td>
<td>61.340</td>
</tr>
<tr>
<td></td>
<td>64.640</td>
<td>64.220</td>
<td>64.220</td>
<td>63.600</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>$\epsilon = 3$</td>
<td>$\epsilon = 8$</td>
<td>$\epsilon = \infty$ (non-DP)</td>
<td>$\epsilon = 3$</td>
</tr>
<tr>
<td></td>
<td>full AUTO-S</td>
<td>full AUTO-S</td>
<td>full AUTO-S</td>
<td>full AUTO-S</td>
</tr>
<tr>
<td></td>
<td>67.857</td>
<td>67.071</td>
<td>67.071</td>
<td>65.872</td>
</tr>
<tr>
<td></td>
<td>68.968</td>
<td>67.533</td>
<td>67.533</td>
<td>66.429</td>
</tr>
</tbody>
</table>

Table 4: Test performance on E2E dataset with GPT2. Additional performance measures are included in Table 8. The best two GPT2 models for each row are marked in bold.

<table>
<thead>
<tr>
<th>Metric</th>
<th>GPT2</th>
<th>GPT2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>full</td>
<td>full</td>
</tr>
<tr>
<td></td>
<td>AUTO-S</td>
<td>AUTO-S</td>
</tr>
<tr>
<td>$\epsilon = 3$</td>
<td>65.872</td>
<td>65.707</td>
</tr>
<tr>
<td>$\epsilon = 8$</td>
<td>66.429</td>
<td>66.429</td>
</tr>
<tr>
<td>$\epsilon = \infty$ (non-DP)</td>
<td>66.429</td>
<td>66.429</td>
</tr>
</tbody>
</table>

Competitive methods include low-rank adaption (LoRA), prefix-tuning (prefix), RGP, only fine-tuning the top 2 Transformer blocks (top2), and training from scratch (retrain), as were recorded in [40]. Again, we use the exactly the same hyperparameters as in [40]. For GPT2 (124 million parameters), GPT2 medium (355 million), and GPT2 large (774 million), Table 4 shows that AUTO-S is scalable with stronger performance on larger models. Our automatic full-parameter finetuning has the best overall performance. Additionally, we highlight that AUTO-S and methods like LoRA are not mutually exclusive and can be combined to yield strong performance, since AUTO-S modifies the optimizers and LoRA modifies the architecture.

7 Discussion

In this work, we proposed the automatic clipping as a drop-in replacement to the standard per-example clipping differentially private training. This is the first technique that eliminate the need to tune the clipping threshold $R$, thus making DP deep learning as easy as regular learning. Our AUTO-S method enjoys both theoretical guarantee of convergence in non-convex problems (under various conditions), and strong empirical performance that advances the state-of-the-art (SOTA) of DP learning on both computer vision and language tasks.

We are excited about the future of automatic DP training, especially along with other working techniques. Notably, our automatic clipping applies compatibly with general optimizers (e.g. [8, 20]), clipping styles (all-layer or per-layer), architecture modifications (e.g. LoRA, RGP, prefix), and data augmentation (e.g. adversarial training [28] and multiple augmentation [16]). Thus, we expect to achieve comparable results to all SOTA in a lightweight fashion.
References


A Proof of differential privacy

Proof of Theorem 3. Define the $\ell_2$ sensitivity of any function $g$ to be $\Delta g = \sup_{S, S'} \|g(S) - g(S')\|_2$ where the supreme is over all neighboring $(S, S')$. Then the Gaussian mechanism $\hat{g}(S) = g(S) + \sigma \Delta g \cdot N(0, I)$.

$\sigma$ denotes the “Noise multiplier”, which corresponds to the noise-level when a Gaussian mechanism is applied to a query with sensitivity $1$.

Observe that automatic clipping (AUTO-V and AUTO-S (4.1)) ensures the bounded global-sensitivity of the stochastic gradient as in Abadi’s clipping. Aligning the noise-multiplier (rather than the noise-level itself) ensures that the noise-to-sensitivity ratio $\sigma \Delta g$ is fixed regardless of $\Delta g$.

The Gaussian mechanism’s privacy guarantees are equivalent. Thus from the privacy accountant perspective, DP-SGD with both Abadi’s clipping and our autoclipping method can be equivalently represented as the adaptive composition of $T$ Poisson sampled Gaussian Mechanism with sampling probability $B/n$ and noise multiplier $\sigma$.

B Proof of automaticity

B.1 Non-adaptive DP optimizers

Proof of Theorem 1. We prove Theorem 1 by showing that, DP-SGD using $R$-dependent AUTO-S with learning rate $\eta$ and weight decay $\lambda$ is equivalent to $R$-independent AUTO-S with learning rate $\eta R$ and weight decay $\lambda / R$. We claim other non-adaptive optimizers such as HeavyBall and NAG can be easily shown in a similar manner.

Recall the standard SGD with weight decay is

$$w_{t+1} = w_t - \eta \left( \sum_{i \in B_t} \frac{\partial l_i}{\partial w_t} + \lambda w_t \right)$$

Replacing the standard gradient $\sum_i \frac{\partial l_i}{\partial w_t}$ with the private gradient, we write the $R$-dependent case as

$$w_{t+1} = w_t - \eta \left( \sum_{i \in B_t} \frac{\partial l_i}{\partial w_t} \cdot \frac{R}{\| \frac{\partial l_i}{\partial w_t} \|_2} + \sigma R \cdot N(0, I) + \lambda w_t \right)$$

$$= w_t - \eta R \left( \sum_{i \in B_t} \frac{\partial l_i}{\partial w_t} / \| \frac{\partial l_i}{\partial w_t} \|_2 + \sigma \cdot N(0, I) \right) - \eta \lambda w_t$$

which is clearly equivalent to the $R$-independent case:

$$w_{t+1} = w_t - \eta' \left( \sum_{i \in B_t} \frac{\partial l_i}{\partial w_t} / \| \frac{\partial l_i}{\partial w_t} \|_2 + \sigma \cdot N(0, I) + \lambda' w_t \right)$$

if we use $\eta' = \eta R$ and $\lambda' = \lambda / R$. □

B.2 Adaptive DP optimizers

Proof of Theorem 2. We prove Theorem 2 by showing that, DP-AdamW using $R$-dependent AUTO-S with learning rate $\eta$ and weight decay $\lambda$ is equivalent to $R$-independent AUTO-S with the same learning rate $\eta$ and weight decay $\lambda / R$. This is the most complicated case. We claim other adaptive optimizers such as AdaDelta, Adam with weight decay (not AdamW), and NAdam can be easily shown in a similar manner.

Recall the standard AdamW is

$$w_{t+1} = w_t - \eta \left( \frac{m_t}{\sqrt{v_t} / (1 - \beta_2) + \lambda w_t} \right)$$
where $\beta_1, \beta_2$ are constants, $g_t := \sum_i \frac{\partial}{\partial w_i}$ is the standard gradient,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \rightarrow m_t = \sum \beta_1^{t-\tau} (1 - \beta_1) g_\tau,$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g^2_t \rightarrow v_t = \sum \beta_2^{t-\tau} (1 - \beta_2) g^2_\tau.$$

Replacing the standard gradient with the private gradient $R\tilde{g}_t := R(\sum_i \frac{\partial}{\partial w_i} | \frac{\partial}{\partial w_i} \|_2 + \sigma \cdot \mathcal{N}(0, I))$, we write the $R$-dependent DP-AdamW as

$$w_{t+1} = w_t - \eta \left( \frac{\tilde{m}_t/(1 - \beta_1)}{\sqrt{v_t/(1 - \beta_2)}} + \lambda w_t \right)$$

where

$$\tilde{m}_t = \beta_1 \tilde{m}_{t-1} + (1 - \beta_1) R\tilde{g}_t \rightarrow \tilde{m}_t = \sum \beta_1^{t-\tau} (1 - \beta_1) R\tilde{g}_\tau,$$

$$\tilde{v}_t = \beta_2 \tilde{v}_{t-1} + (1 - \beta_2) R^2\tilde{g}^2_t \rightarrow \tilde{v}_t = \sum \beta_2^{t-\tau} (1 - \beta_2) R^2\tilde{g}^2_\tau.$$

Clearly, the $R$ factor in the numerator and denominator of $\frac{\tilde{m}_t/(1 - \beta_1)}{\sqrt{v_t/(1 - \beta_2)}}$ cancel each other. Therefore we claim that the $R$-dependent DP-AdamW is in fact completely independent of $R$. \hfill \Box

### B.3 Automatic per-layer clipping

In some cases, the per-layer clipping is desired, where we use a clipping threshold vector $\mathbf{R} = [R_1, \ldots, R_L]$ and each layer uses a different clipping threshold. We claim that DP optimizers under automatic clipping works with the per-layer clipping when $\mathbf{R}$ is tuned proportionally, e.g. $\mathbf{R} = R \cdot [a_1, \ldots, a_L]$, but not entry-wise (see counter-example in Fact B.1). One special case is the uniform per-layer clipping when $R_1 = \cdots = R_L = R/\sqrt{L}$. This is widely applied as only one norm $R$ requires tuning, instead of $L$ norms in $\mathbf{R}$, particularly in the case of deep models with hundreds of layers. The corresponding DP-SGD with AUTO-S in (3.3) gives

$$w^{(l)}_{t+1} = w^{(l)}_t - \eta \left( \sum_{i \in B_l} \frac{R}{\sqrt{L} \| g^{(l)}_{i,t,i} \| + \gamma} \cdot g^{(l)}_{i,t,i} + \sigma R \cdot \mathcal{N}(0, I) \right)$$

Here the superscript $(l)$ is the layer index. Clearly $R$ couples with the learning rate $\eta$ and the same analysis as in Theorem 1 follows. The adaptive optimizers can be similarly analyzed from Theorem 2.

**Fact B.1.** Changing one clipping threshold in the clipping threshold vector $\mathbf{R}$ (i.e. not proportionally) can break the coupling with learning rate.

**Proof of Fact B.1.** We prove by a counter-example of $\mathbf{R}$ in $\mathbb{R}^2$. Consider DP-SGD with per-layer clipping thresholds $(R_1, R_2) = (9, 12)$:

$$w^{(l)}_{t+1} = w^{(l)}_t - \eta \left( \sum_{i \in B} \frac{R_1 g^{(l)}_{i,t,i}}{\| g^{(l)}_{i,t,i} \|} + \sqrt{R_1^2 + R_2^2} \cdot \mathcal{N}(0, I) \right)$$

Increasing $R_1$ from 9 to 16 changes the update for the first layer

$$\eta \left( \sum_{i \in B} \frac{9g^{(l)}_{i,t,i}}{\| g^{(l)}_{i,t,i} \|} + 15\sigma \cdot \mathcal{N}(0, 1) \right) \rightarrow \eta \left( \sum_{i \in B} \frac{16g^{(l)}_{i,t,i}}{\| g^{(l)}_{i,t,i} \|} + 20\sigma \cdot \mathcal{N}(0, 1) \right)$$

The noise-to-signal ratio decreases from $5/3$ to $5/4$ for this layer, and increases from $5/4$ to $5/3$ for the second layer. This breaks the coupling with learning rate, since the coupling does not change the noise-to-signal ratio. \hfill \Box
C Main results of convergence for DP-SGD with automatic clipping

C.1 Main proof of convergence for DP-SGD (the envelope version)

Proof of Theorem 4. In this section, we prove two parts of Theorem 4. The first part of Theorem 4 is the upper bound on $\min_t E(\|g_t\|)$, which is a direct result following from Theorem 6, and we prove it in Appendix C.2.

Theorem 6. Under Assumption 5.1, 5.2, 5.3, running DP-SGD with automatic clipping for $T$ iterations gives

$$\min_t E(\|g_t\|) \leq \frac{\xi}{r} + F \left( \frac{4}{\sqrt{T}} \sqrt{\left( \mathcal{L}_0 - \mathcal{L}_* \right) \left( 1 + \frac{\sigma^2 d}{B^2} \right) r; \xi, \gamma} \right) \ (C.1)$$

where

- for $r < 1$, $\gamma = 0$ and $\eta \propto 1/\sqrt{T}$, $F(x) = \min_{0 < c < 1} f(c, r)$ and $f(c, r) := \frac{(1+cr)}{\sqrt{r^2 - 2rc + 1}} + \frac{(1-rc)}{\sqrt{r^2 - 2rc + 1}}$: for $r \geq 1$, $\gamma = 0$ and $\eta \propto 1/\sqrt{T}$, $F(x) = \infty$;

- for $r \geq 1$, $\gamma > 0$ and $\eta \propto 1/\sqrt{T}$, $F$ is the convex envelope of (C.8), and is strictly increasing.

![Figure 6: Visualization of upper bound $\frac{\xi}{r} + F \left( O(1/\sqrt{T}); r, \xi, \gamma \right)$ for gradient norm, with $O(1/\sqrt{T})$ in (C.1). Here $\xi = 1$. The right plot is a zoom-in (with additional lines) of the left one.](image)

Notice that, (C.1) holds for any $r > 0$. However, we have to consider an envelope curve over $r$ in (C.1) to reduce the upper bound: with AUTO-V clipping ($\gamma = 0$), the upper bound in (C.1) is always larger than $\xi$ as $r < 1$; we must use AUTO-S clipping ($\gamma > 0$) to reduce the upper bound to zero, as can be seen from Figure 6. In fact, larger $T$ needs larger $r$ to reduce the upper bound.

All in all, we specifically focus on $r \geq 1$ and $\gamma > 0$, which is the only scenario that (C.1) can converge to zero. This scenario is also where we prove the second part of Theorem 4.

The second part of Theorem 4 is the asymptotic convergence rate $O(T^{-1/4})$ of DP-SGD, only possible under $r \geq 1$ and $\gamma > 0$.

By (C.1) in Theorem 6, our upper bound $G$ from Theorem 4 can be simplified to

$$\min_{r > 0} \frac{\xi}{r} + (M^{-1})_{ccv} \left( \frac{4}{\sqrt{T}} \sqrt{\left( \mathcal{L}_0 - \mathcal{L}_* \right) \left( 1 + \frac{\sigma^2 d}{B^2} \right) r; \xi, \gamma} \right)$$

where the function $M^{-1}$ is explicitly defined in (C.8) and the subscript $ccv$ means the upper concave envelope. Clearly, as $T \to \infty$, $M^{-1}(\frac{1}{\sqrt{T}}) \to 0$. We will next show that the convergence rate of $M^{-1}$ is indeed $O(\frac{1}{\sqrt{T}})$ and the minimization over $r$ makes the overall convergence rate $O(T^{-1/4})$. 

17
Starting from (C.8), we denote \( x = \frac{4}{\sqrt{T}} \sqrt{(L_0 - L_*) L \left( 1 + \frac{\sigma^2 L^2}{r^2} \right)} \) and write

\[
\mathcal{M}^{-1}(x; r, \xi, \gamma) = \frac{-\xi r + (r^2 - 1) \xi x + r \gamma x + \gamma \sqrt{(\xi r)^2 + 2 \xi x + 2 \gamma x + x^2}}{2 \gamma - (r^2 - 1) x} \\
= \left( -\frac{\xi}{r} + (r^2 - 1) \frac{\xi}{r} x + r \gamma x + \gamma \sqrt{(\frac{\xi}{r})^2 + 2 \xi x + 2 \gamma x + x^2} \right) \\
\cdot \left( 1 + \frac{r^2 - 1}{2 \gamma} x + O(x^2) \right) \\
= \frac{1}{2 \gamma} \left( -\frac{\xi}{r} + (r^2 - 1) \frac{\xi}{r} x + r \gamma x + \frac{\gamma \xi}{r} \sqrt{1 + \frac{2 (\xi + \gamma) r^2 x}{\xi^2} + O(x^2)} \right) \\
\cdot \left( 1 + \frac{r^2 - 1}{2 \gamma} x + O(x^2) \right) \\
= \frac{1}{2 \gamma} \left( (r^2 - 1) \frac{\xi}{r} x + r \gamma x + \frac{\gamma (\xi + \gamma) r x}{\xi} + O(x^2) \right) \cdot \left( 1 + \frac{r^2 - 1}{2 \gamma} x + O(x^2) \right) \\
= \frac{1}{2 \gamma} \left( (r^2 - 1) \frac{\xi}{r} x + r \gamma x + \frac{\gamma (\xi + \gamma) r x}{\xi} \right) \cdot x + O(x^2) \\
= \frac{1}{2 \gamma} \left( \frac{(\xi + \gamma)^2}{\xi} r - \frac{\xi}{r} \right) \cdot x + O(x^2)
\]

Since \( \mathcal{M}^{-1} \) is asymptotically linear as \( x \to 0 \), we instead study

\[
\min_{r > 0} \frac{\xi}{r} + \mathcal{M}^{-1}(x; r, \xi, \gamma) \equiv \min_{r > 0} \frac{\xi}{r} + \frac{1}{2 \gamma} \left( \frac{(\xi + \gamma)^2}{\xi} r - \frac{\xi}{r} \right) \cdot x + O(x^2).
\]

That is, ignoring the higher order term for the asymptotic analysis, the \( \mathcal{M}^{-1} \) part converges as \( O(x) = O(1/\sqrt{T}) \), and we visualize this in Figure 8.

Although DP-SGD converges faster than SGD, the former converges to \( \xi/r \) and the latter converges to 0. Thus, taking \( \xi/r \) into consideration, the objective reduces to a hyperbola

\[
\frac{(\xi (1 - \frac{\xi}{2 \gamma}))}{r} + \frac{x(\xi + \gamma)^2}{2 \gamma \xi} \cdot r
\]

whose minimum over \( r \) is obviously

\[
2 \sqrt{\xi \left( 1 - \frac{x}{2 \gamma} \right) \frac{\gamma (\xi + \gamma)^2}{2 \gamma \xi}} = O(\sqrt{x}) = O(T^{-1/4}).
\]

To give more details about the upper bound in (5.2), we demonstrate its dependence on \( \xi \) and \( \gamma \) in Figure 7.

### C.2 Main proof of convergence for DP-SGD (the non-envelope version)

**Proof of Theorem 6.** Consider DP-SGD with AUTO-S clipping

\[
w_{t+1} = w_t - \eta \left( \sum_i \frac{\tilde{g}_{t,i}}{\|\tilde{g}_{t,i}\| + \gamma} + \sigma N(0, I) \right)
\]

where \( \tilde{g}_{t,i} \) is i.i.d. samples of \( \tilde{g}_t \), an unbiased estimate of \( g_t \), with a bounded variance as described in Assumption 5.3.
Figure 7: Dependence of the upper bound $G$ on $\xi$ (left) and $\gamma$ (right). Here the $O(1/\sqrt{T})$ term is set to 10 and either $\gamma = 0$ (left) or $\xi = 1$ (right).

Figure 8: Convergence with respect to $T$. Same setting as Figure 5.

By Lipschitz smoothness in Assumption 5.2, and denoting $Z = N(0, I)$, we have

$$
\mathcal{L}_{t+1} - \mathcal{L}_t \leq g_t^\top (w_{t+1} - w_t) + \frac{L}{2} \| w_{t+1} - w_t \|^2
$$

$$
= -\eta g_t^\top \left( \sum_i \frac{\tilde{g}_{t,i}}{\|\tilde{g}_{t,i}\| + \gamma} + \sigma Z \right) + \frac{L \eta^2}{2} \left( \sum_i \frac{\tilde{g}_{t,i}}{\|\tilde{g}_{t,i}\| + \gamma} + \sigma Z \right)^2
$$

$$
\leq -\eta g_t^\top \left( \sum_i \frac{\tilde{g}_{t,i}}{\|\tilde{g}_{t,i}\| + \gamma} + \sigma Z \right)

+ L \eta^2 \left( \left\| \sum_i \frac{\tilde{g}_{t,i}}{\|\tilde{g}_{t,i}\| + \gamma} \right\|^2 + \sigma^2 \|Z\|^2 \right)
$$

where the last inequality follows from Cauchy Schwartz.

Given the fact that $\|\tilde{g}_{t,i}/(\|\tilde{g}_{t,i}\| + \gamma)\| \leq 1$, the expected improvement at one iteration is

$$
\mathbb{E}(\mathcal{L}_{t+1} - \mathcal{L}_t | w_t) \leq -\eta g_t^\top \mathbb{E} \left( \sum_i \frac{\tilde{g}_{t,i}}{\|\tilde{g}_{t,i}\| + \gamma} \right) + L \eta^2 (B^2 + \sigma^2 d)
$$

$$
= -\eta B g_t^\top \mathbb{E} \left( \frac{\tilde{g}_t}{\|\tilde{g}_t\| + \gamma} \right) + L \eta^2 (B^2 + \sigma^2 d)
$$

(C.2)

Now we want to lower bound $g_t^\top \mathbb{E} \left( \frac{\tilde{g}_t}{\|\tilde{g}_t\| + \gamma} \right)$ in (C.2).
Write \( \hat{g}_t = g_t + \Delta_t \) where the gradient noise \( \Delta_t \) follows \( \mathbb{E}\Delta_t = 0, \mathbb{E}\|\Delta_t\| < \xi \) by Assumption 5.3. Then

\[
\mathbb{E} \left( \frac{\hat{g}_t}{\|\hat{g}_t\| + \gamma} \right) = \mathbb{E} \left( \frac{\|g_t\|^2 + g_t^\top \Delta_t}{\|g_t + \Delta_t\| + \gamma} \right) = \frac{1}{2} \mathbb{E} \left( \frac{\|g_t\|^2 + g_t^\top \Delta_t}{\|g_t + \Delta_t\| + \gamma} \mid \Delta_t \in H_+ \right) + \frac{1}{2} \mathbb{E} \left( \frac{\|g_t\|^2 - g_t^\top \Delta_t}{\|g_t - \Delta_t\| + \gamma} \mid \Delta_t \in H_- \right)
\]

where we use the hyperplane perpendicular to \( g_t \) to divide the support of \( \Delta_t \) into two half-spaces:

\[
H_+ := \{ v : g_t^\top v > 0 \}, \quad H_- := \{ v : g_t^\top v < 0 \}.
\]

We use the symmetry assumption in Assumption 5.3 to get

\[
\mathbb{P}(\Delta_t \in H_+) = \mathbb{P}(\Delta_t \in H_-) = \frac{1}{2}
\]

and notice that \( \Delta_t \overset{d}{=} -\Delta_t \), i.e., if \( \Delta_t \in H_+ \), then \( -\Delta_t \in H_- \) with the same distribution.

The next result further gives a lower bound for \( \mathbb{E} \left( \frac{\hat{g}_t}{\|\hat{g}_t\| + \gamma} \right) \) using \( \|g_t\| \).

**Lemma C.1.**

\[
\mathbb{E} \left( \frac{\|g_t\|^2 + g_t^\top \Delta_t}{\|g_t + \Delta_t\| + \gamma} + \frac{\|g_t\|^2 - g_t^\top \Delta_t}{\|g_t - \Delta_t\| + \gamma} \mid \Delta_t \in H_+ \right) \geq \min_{0 < c < 1} f(c, r; \gamma) \cdot (\|g_t\| - \xi/r)
\]

for any \( r > 0 \) and \( f(c, r; \Gamma) = \frac{(1+c)\sqrt{2c+1}}{\sqrt{r^2+2c+1}} \). For the simplicity of notation, we denote the distance measure

\[
\mathcal{M}(\|g_t\| - \xi/r; r, \xi, \gamma) = \min_{0 < c < 1} f(c, r; \gamma) \cdot (\|g_t\| - \xi/r)
\]

and leave the fine-grained analysis (e.g. its explicit form in some scenarios) at the end of this section. Using the lower bound from Lemma C.1, the expected improvement (C.2) becomes

\[
\mathbb{E}(L_{t+1} - L_t | w_t) \leq -\frac{\eta B}{2} \mathcal{M}(\|g_t\| - \xi/r) + L\eta^2 B^2 \left( 1 + \frac{\sigma^2 d}{B^2} \right)
\]

Now extend the expectation over randomness in the trajectory, and perform a telescoping sum over the iterations

\[
L_0 - L_* \geq L_0 - \mathbb{E}L_T = \sum_t \mathbb{E}(L_t - L_{t+1}) \geq \frac{\eta B}{2} \mathbb{E} \left( \sum_t \mathcal{M}(\|g_t\| - \xi/r) \right) - TL\eta^2 B^2 \left( 1 + \frac{\sigma^2 d}{B^2} \right)
\]

Substituting \( \eta B = \eta_0 / \sqrt{T} \) where \( \eta_0 \) is a base learning rate, we have

\[
2(L_0 - L_*) \geq \sqrt{T} \eta_0 \mathbb{E} \left( \frac{1}{T} \sum_t \mathcal{M}(\|g_t\| - \xi/r) \right) - 2L\eta_0^2 \left( 1 + \frac{\sigma^2 d}{B^2} \right)
\]

and finally

\[
\mathbb{E} \left( \frac{1}{T} \sum_t \mathcal{M}(\|g_t\| - \xi/r) \right) \leq \frac{1}{\sqrt{T}} \left[ \frac{2(L_0 - L_*)}{\eta_0} + 2L\eta_0 \left( 1 + \frac{\sigma^2 d}{B^2} \right) \right]
\]

(C.4)
With $\eta_0$ chosen properly at $\eta_0 = \sqrt{\frac{\mathcal{L}_0 - \mathcal{L}_*}{L \left(1 + \frac{\sigma^2 d}{B^2}\right)}}$, the hyperbola on the right hand side in (C.4) is minimized to $4\sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)}$, and we obtain
\[
\mathbb{E}\left(\frac{1}{T} \sum_t \mathcal{M}(||g_t|| - \xi/r)\right) \leq \frac{4}{\sqrt{T}} \sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)}
\]
Since the minimum of a sequence is smaller than the average, we have
\[
\min_t \mathbb{E}(\mathcal{M}(||g_t|| - \xi/r)) \leq \frac{1}{T} \sum_t \mathbb{E}(\mathcal{M}(||g_t|| - \xi/r)) \leq \frac{4}{\sqrt{T}} \sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)} \tag{C.5}
\]
We claim that $\mathcal{M}$ may not be concave or convex. Therefore we use $\mathcal{M}_{cvx}$ to denote its lower convex envelope, i.e. the largest convex function that is smaller than $\mathcal{M}$. Then by Jensen’s inequality (C.5) becomes
\[
\min_t \mathcal{M}_{cvx}(\mathbb{E}(||g_t|| - \xi/r)) \leq \min_t \mathbb{E}(\mathcal{M}_{cvx}(||g_t|| - \xi/r)) \leq \frac{4}{\sqrt{T}} \sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)} \tag{C.6}
\]
It is obvious that $\mathcal{M}_{cvx}$ is increasing as $\mathcal{M}$ is increasing by Theorem 8. Hence, $(\mathcal{M}_{cvx})^{-1}$ is also increasing, as the inverse of $\mathcal{M}_{cvx}$. We write (C.6) as
\[
\min_t \mathbb{E}(||g_t|| - \xi/r) \leq (\mathcal{M}_{cvx})^{-1} \left(\frac{4}{\sqrt{T}} \sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)}\right)
\]
and equivalently
\[
\min_t \mathbb{E}(||g_t||) \leq \frac{\xi}{r} + (\mathcal{M}_{cvx})^{-1} \left(\frac{4}{\sqrt{T}} \sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)}\right) \tag{C.7}
\]
Finally, we derive the explicit properties of $\mathcal{M}(||g_t|| - \xi/r)$ in Theorem 8. These properties allow us to further analyze on the convergence of $\mathcal{M}(||g_t|| - \xi/r)$, based on AUTO-V and AUTO-S, respectively.

1. **DP-SGD with AUTO-V clipping.** By Theorem 8, we write
\[
\mathcal{M}(x; r) = \min_{c \in [0, 1]} f(c, r; 0) \cdot x
\]
This is a linear function and thus $\mathcal{M}_{cvx} = \mathcal{M} = 1/\mathcal{M}_{cvx}^{-1}$. As a result, we have
\[
\min_t \mathbb{E}(||g_t||) \leq \frac{\xi}{r} + \frac{1}{\min_{c \in [0, 1]} f(c, r; 0)} \cdot \frac{4}{\sqrt{T}} \sqrt{(\mathcal{L}_0 - \mathcal{L}_*)L \left(1 + \frac{\sigma^2 d}{B^2}\right)}
\]
We note here $r$ plays an important role under AUTO-V clipping: when $r < 1$, we spend more iterations to converge to better and smaller gradient norm $\xi/r$; when $r \geq 1$, $\min_c f(c, r; 0) = f(1, r; 0) = 0$ and it takes forever to converge. This is demonstrated in the left plot of Figure 5.

2. **DP-SGD with AUTO-S clipping.** By Theorem 8 and for $r > 1$, we write
\[
\mathcal{M}(x; r, \xi, \gamma) = \left(\frac{\gamma}{(r - 1)(x + \xi/r) + \gamma} - \frac{\gamma}{(r + 1)(x + \xi/r) + \gamma}\right) \cdot x.
\]
Notice that the inverse of a lower convex envelope is equivalent to the upper concave envelope (denoted by the subscript $ccv$) of an inverse. Therefore we can derive $(\mathcal{M}_{cvx})^{-1} = (\mathcal{M}^{-1})_{ccv}$ with the explicit form
\[
\mathcal{M}^{-1}(x; r, \xi, \gamma) = \frac{-\frac{\xi}{r} \gamma + (r^2 - 1)\xi + r \gamma + \gamma \sqrt{(\frac{\xi}{r})^2 + 2\xi + 2\gamma x + x^2}}{2\gamma - (r^2 - 1)x} \tag{C.8}
\]
we can derive it based on \( r, \xi, \gamma \) and substitute back to (C.7).

Note that the domain of \( M^{-1} \) (or the image of \( M \)) is \([0, \frac{\gamma}{r-1} - \frac{\gamma}{r+1}]\).

In comparison to the AUTO-V clipping, \( M^{-1} \) takes a much more complicated form, as depicted in the middle plot of Figure 5, where \( r > 1 \) plays an important role for the gradient norm to converge to zero.

### C.3 Proof of Lemma C.1

**Proof of Lemma C.1.** We want to lower bound

\[
\mathbb{E} \left( \frac{\|g_t\|^2 + g_t^\top \Delta_t}{\|g_t + \Delta_t\| + \gamma} + \frac{\|g_t\|^2 - g_t^\top \Delta_t}{\|g_t - \Delta_t\| + \gamma} \mid \Delta_t \in H_+ \right) \tag{C.9}
\]

To simplify the notation, we denote noise-to-signal ratio \( S := \frac{\|\Delta_t\|}{\|g_t\|} \) and \( c := \cos \theta = \frac{g_t^\top \Delta_t}{\|g_t\| \|\Delta_t\|} \), with \( \theta \) be the random angle between \( g_t \) and \( \Delta_t \). Note that \( 0 < c \leq 1 \) when \( \Delta_t \in H_+ \).

The term inside the conditional expectation in (C.9) can be written as

\[
\frac{(1 + Sc)\|g_t\|^2}{\sqrt{S^2 + 2Sc + 1}\|g_t\| + \gamma} + \frac{(1 - Sc)\|g_t\|^2}{\sqrt{S^2 - 2Sc + 1}\|g_t\| + \gamma} = \|g_t\| \left( \frac{(1 + Sc)}{\sqrt{S^2 + 2Sc + 1} + \gamma/\|g_t\|} + \frac{(1 - Sc)}{\sqrt{S^2 - 2Sc + 1} + \gamma/\|g_t\|} \right).
\]

Defining \( \Gamma = \gamma/\|g_t\| \) and

\[
f(c, S; \Gamma) := \frac{(1 + Sc)}{\sqrt{S^2 + 2Sc + 1} + \Gamma} + \frac{(1 - Sc)}{\sqrt{S^2 - 2Sc + 1} + \Gamma}, \tag{C.10}
\]

we turn the conditional expectation in (C.9) into

\[
\mathbb{E} \left( \frac{\|g_t\|^2 + g_t^\top \Delta_t}{\|g_t + \Delta_t\| + \gamma} + \frac{\|g_t\|^2 - g_t^\top \Delta_t}{\|g_t - \Delta_t\| + \gamma} \mid \Delta_t \in H_+ \right) = \|g_t\| \mathbb{E}(f(c, S; \Gamma) \mid \Delta_t \in H_+) \tag{C.11}
\]

for which we want to lower bound \( f(c, S; \Gamma) \) over \( 0 < c \leq 1, S > 0, \Gamma > 0 \). We use the next theorem to prepare some helpful properties. The proof can be found in Appendix E.1.

**Theorem 7.** For \( f \) defined in (C.10), we have

1. \( f(c, S; \Gamma) \) is strictly decreasing in \( S \) for all \( 0 < c < 1 \) and \( \Gamma > 0 \).
2. Consequently, \( \min_{c \in (0, 1)} f(c, S; \Gamma) \) is strictly decreasing in \( S \).
3. \( f(c, S; \Gamma) \) is strictly decreasing in \( c \) for all \( S > 1 \) and \( \Gamma > 0 \).

We consider a thresholding ratio \( r > 0 \) and we will focus on the regime that \( S < r \). This \( r \) will turn out to measure the minimum gradient norm at convergence: informally speaking, \( \|g_t\| \) converges to \( \xi/r \).
By the law of total expectation, (C.11) can be relaxed as follows.
\[
\|g_t\|E \left( f(c, S; \Gamma) \middle| \Delta \in H_+ \right) \\
\geq \|g_t\|E \left( f(c, S; \Gamma) \middle| \Delta \in H_+, S < r \right) \mathbb{P}(r\|g_t\| > \|\Delta\| \middle| \Delta \in H_+) \\
\quad + \|g_t\|E \left( f(c, S; \Gamma) \middle| \Delta \in H_+, S > r \right) \mathbb{P}(r\|g_t\| < \|\Delta\| \middle| \Delta \in H_+) \\
\geq \|g_t\|E \left( f(c, r; \Gamma) \middle| \Delta \in H_+, S < r \right) \mathbb{P}(r\|g_t\| > \|\Delta\| \middle| \Delta \in H_+) \\
\geq \|g_t\|E \left( f(c, r; \Gamma) \middle| \Delta \in H_+, S < r \right) \mathbb{P}(r\|g_t\| > \|\Delta\|) \\
\geq \min_{c \in (0,1]} f(c, r; \Gamma) \cdot \frac{\|g_t\|\mathbb{P}(r\|g_t\| > \|\Delta\|)}{\|\Delta\|} \\
\tag{C.12}
\]
where in the first inequality, the ignoring of last term is justified by \( f(c, S; \Gamma) \geq \min_{c \in (0,1]} f(c, S; \Gamma) \geq \min_{c \in (0,1]} f(c, \infty; \Gamma) = 0 \), from the monotonicity (second statement) in Theorem 7.

We first lower bound (\( \circ \)) by applying the Markov’s inequality:
\[
\mathbb{P}(r\|g_t\| > \|\Delta_t\|) \geq 1 - \frac{\mathbb{E}[\|\Delta_t\|]}{r\|g_t\|}
\]
and hence by Assumption 5.3,
\[
\|g_t\|\mathbb{P}(r\|g_t\| > \|\Delta_t\|) \geq \|g_t\| - \mathbb{E}[\|\Delta_t\|]/r \geq \|g_t\| - \xi/r.
\]
Finally, the conditional expectation of interest in (C.9) gives
\[
\mathbb{E}\left( \frac{\|g_t\|^2 + g_t^\top \Delta_t}{\|g_t\| + \|\Delta_t\|} + \frac{\|g_t\|^2 - g_t^\top \Delta_t}{\|g_t - \Delta_t\|} \middle| \Delta_t \in H_+ \right) \geq \min_{0 < c \leq 1} f(c, r; \frac{\gamma}{\|g_t\|}) \cdot (\|g_t\| - \xi/r)
\]
\[\Box\]

C.4 Proof of Theorem 8

To derive some properties of \( \min_c f(c, r; \Gamma) \), we need to compute separately for AUTO-V (without the stability constant, \( \Gamma = 0 \)) and for AUTO-S (with the stability constant, \( \Gamma > 0 \)), as shown in Theorem 8. As we will show, as the number of training iterations \( T \to \infty \), DP-SGD with AUTO-V clipping can only compress \( \|g_t\| \) to \( \xi/r \) for \( r < 1 \). However, DP-SGD with AUTO-S clipping can compress \( \|g_t\| \) to \( \xi/r \) for any \( r > 1 \).

Theorem 8.

1. For \( 0 < r < 1 \) and \( \Gamma = 0 \), we have \( \min_{c \in (0,1]} f(c, r; 0) > 0 \). Then Equation (C.11) is lower bounded by
\[
\min_{c \in (0,1]} f(c, r; 0) \cdot (\|g_t\| - \xi/r)
\]
which is increasing in \( \|g_t\| - \xi/r \).

2. For \( r \geq 1 \) and \( \Gamma = 0 \), we have \( \min_{c \in (0,1]} f(c, r; \Gamma) = f(1, r; 0) = 0 \). In words, (C.9) has a trivial lower bound and Theorem 6 cannot compress \( \|g_t\| \) to \( \xi/r \).

3. For \( r \geq 1 \) and \( \Gamma > 0 \), we have \( \min_{c \in (0,1]} f(c, r; \Gamma) = f(1, r; \Gamma) = \left( \frac{\Gamma}{r + 1 - r} - \frac{\Gamma}{r + 1 + 1} \right) \). Then Equation (C.11) is lower bounded by
\[
\left( \frac{\gamma}{(r - 1)\|g_t\| + \gamma} - \frac{\gamma}{(r + 1)\|g_t\| + \gamma} \right) \cdot (\|g_t\| - \xi/r)
\]
which is increasing in \( \|g_t\| - \xi/r \).

Proof. To prove statement 1, we use the second statement from Theorem 7 and show that \( \min_c f(c, r; 0) > \min_c f(c, \infty; 0) = 0 \). To prove statement 2 and 3, we use the third statement from Theorem 7 and see that \( \min_c f(c, r; \Gamma) = f(1, r; \Gamma) \) with an explicit formula. \( \Box \)
D Convergence rate of standard SGD

Theorem 9. Under Assumption 5.1, 5.2, 5.3 (without the symmetry assumption), running the standard non-DP SGD for \( T \) iterations gives, for \( \eta \propto 1/\sqrt{T} \),
\[
\min_t \mathbb{E}(\|g_t\|) \leq \frac{1}{T^{1/4}} \sqrt{2(L_0 - L_\ast)L + \frac{\xi^2}{B}}
\]

Proof of Theorem 9. Consider the standard SGD
\[
w_{t+1} = w_t - \eta \sum_i \tilde{g}_{t,i}
\]
where \( \tilde{g}_{t,i} \) is i.i.d. unbiased estimate of \( g_t \), with a bounded variance as described in Assumption 5.3.

By Lipschitz smoothness assumption in Assumption 5.2,
\[
L_{t+1} - L_t \leq g_t^\top (w_{t+1} - w_t) + \frac{L}{2} \|w_{t+1} - w_t\|^2 = -\eta g_t^\top \left( \sum_i \frac{1}{B} \tilde{g}_{t,i} \right) + \frac{L\eta^2}{2} \left\| \sum_i \frac{1}{B} \tilde{g}_{t,i} \right\|^2
\]
The expected improvement at one iteration is
\[
\mathbb{E}(L_{t+1} - L_t | w_t) \leq -\eta \|g_t\|^2 + \frac{L\eta^2}{2} \left( \|g_t\|^2 + \frac{\xi^2}{B} \right) \tag{D.1}
\]
Now we extend the expectation over randomness in the trajectory, and perform a telescoping sum over the iterations
\[
L_0 - L_\ast \geq L_0 - \mathbb{E}L_T = \sum_t \mathbb{E}(L_t - L_{t+1}) \geq \left( \eta - \frac{L\eta^2}{2} \right) \mathbb{E}(\sum_t \|g_t\|^2) - \frac{T L\eta^2 \xi^2}{2B}
\]
Notice that we do not need the symmetry assumption in Assumption 5.3 in the non-DP SGD analysis.

We apply the same learning rate as in [5], \( \eta = \frac{1}{L\sqrt{T}} \).
\[
2(L_0 - L_\ast) \geq \left( \frac{2}{L \sqrt{T}} - \frac{1}{LT} \right) \mathbb{E} \left( \sum_t \|g_t\|^2 \right) - \frac{T \xi^2}{BL} \geq \frac{1}{T} \sum_t \|g_t\|^2 - \frac{\xi^2}{BL}
\]
and finally
\[
\min_t \mathbb{E}(\|g_t\|^2) \leq \mathbb{E} \left( \frac{1}{T} \sum_t \|g_t\|^2 \right) \leq \frac{1}{\sqrt{T}} \left[ 2(L_0 - L_\ast)L + \frac{\xi^2}{B} \right]
\]
Using the Jensen’s inequality, we can have
\[
\min_t \mathbb{E}(\|g_t\|) \leq \frac{1}{T^{1/4}} \sqrt{2(L_0 - L_\ast)L + \frac{\xi^2}{B}} \tag{E.1}
\]

E Auxiliary proofs

E.1 Proof of Theorem 7

Proof. We first show \( \frac{df(c,S;\Gamma)}{dS} < 0 \) for all \( 0 < c < 1, \Gamma > 0 \) and \( S > 0 \), as visualized in the left plot of Figure 9. We can explicitly write down the derivative, by WolframAlpha
\[
\frac{df(c,S;\Gamma)}{dS} = \frac{-(A\Gamma^2 + B\Gamma + C)}{\sqrt{S^2 - 2cS + 1}\sqrt{S^2 + 2cS + 1}(\Gamma + \sqrt{S^2 - 2cS + 1})^2(\Gamma + \sqrt{S^2 + 2cS + 1})^2}
\]
with
\[ A(c, S) = \sqrt{S^2 + 2cS + 1} \left( 3c^2 S - 2c(S^2 + 1) + S \right) + \sqrt{S^2 - 2cS + 1} \left( 3c^2 S + 2c(S^2 + 1) + S \right) \]
\[ B(c, S) = 4S \left( (S^2 + 1)(1 - c^2) + c^2 \sqrt{S^2 + 2cS + 1} \sqrt{S^2 - 2cS + 1} \right) \]
\[ C(c, S) = (1 - c^2)S \left[ (S^2 - 2cS + 1)^{3/2} + (S^2 + 2cS + 1)^{3/2} \right] \]

It is obvious that, since \( c < 1 \),
\[ S^2 \pm 2cS + 1 > S^2 \pm 2cS + c^2 = (S \pm c)^2 \geq 0. \] (E.2)

From (E.2), the denominator in (E.1) is positive and it suffices to show \( A^2 + B^2 + C > 0 \) for all \( 0 < c < 1 \) and \( S > 0 \), in order to show \( \frac{df}{dc} < 0 \).

Also from (E.2), we can easily see \( B(c, S) > 0 \) and \( C(c, S) > 0 \). We will show that \( A(c, S) > 0 \) in Lemma E.1, after very heavy algebraic computation.

Now we can claim that \( A^2 + B^2 + C > 0 \) by Fact E.3, and complete the proof of the first statement.

To further see that \( \min_c f(c, S; \Gamma) \) is decreasing in \( S \), let us denote \( c^*(x; \Gamma) := \arg \min_{c \in [0,1]} f(c, x; \Gamma) \). Then considering \( S < S' \), we prove the second statement by observing
\[ \min_c f(c, S; \Gamma) = f(c^*(S; \Gamma), S; \Gamma) > f(c^*(S'; \Gamma), S'; \Gamma) \geq \min_c f(c, S'; \Gamma). \]

This statement is also visualized in the right plot of Figure 9.

We next show \( \frac{df(c, S; \Gamma)}{dc} < 0 \) for all \( 0 < c < 1 \), \( \Gamma > 0 \) and \( S > 1 \). We can explicitly write down the derivative, by WolframAlpha
\[
\frac{df(c, S; \Gamma)}{dc} = \frac{-S(A'T^2 + B'T + C')}{\sqrt{S^2 - 2cS + 1} \sqrt{S^2 + 2cS + 1}(\Gamma + \sqrt{S^2 - 2cS + 1})^2(\Gamma + \sqrt{S^2 + 2cS + 1})^2}
\] (E.3)

with
\[ A'(c, S) = \left[ (S^2 + 3cS + 2) \sqrt{S^2 - 2cS + 1} - (S^2 - 3cS + 2) \sqrt{S^2 + 2cS + 1} \right] \]
\[ B'(c, S) = 4Sc \left[ \sqrt{S^2 + 2cS + 1} \sqrt{S^2 - 2cS + 1} + (S^2 - 1) \right] \]
\[ C'(c, S) = S \left[ (c + S)(S^2 - 2cS + 1)^{3/2} + (c - S)(S^2 + 2cS + 1)^{3/2} \right] \]

Clearly \( B'(c, S) > 0 \) and \( C'(c, S) > 0 \), since \( S^2 + 2cS + 1 > S^2 - 2cS + c^2 = (S - c)^2 \geq 0 \). And we will show \( A'(c, S) > 0 \) in Lemma E.2, after some algebra.

We again claim that \( A'T^2 + B'T + C' > 0 \) by Fact E.3, which guarantees that the numerator in (E.3) is negative and that \( \frac{df}{dc} < 0 \). This is visualized in Figure 10.

\[ \text{Figure 9: Visualization of } f(0.5, S; \Gamma) \text{ (left) and } \min_{0 \leq c \leq 1} f(c, S; \Gamma) \text{ over } S > 0. \]
Figure 10: Visualization of $f(c, 0.8, \Gamma)$ (left) and $f(c, 2, \Gamma)$ over $0 \leq c \leq 1$.

### E.2 Proof of Lemma E.1

**Lemma E.1.** For all $0 < c < 1$ and $S > 0$,  
$$A := \sqrt{S^2 + 2cS + 1 - 3c^2} < \sqrt{S^2 + 2cS + 1 + 3c^2}$$

**Proof.** We prove by contradiction. Suppose
$$\sqrt{S^2 + 2cS + 1 - 3c^2} < 2.$$  
Then
$$0 < \sqrt{S^2 - 2cS + 1 + 3c^2} < -\sqrt{S^2 + 2cS + 1 + 3c^2}.$$  
where the first inequality comes from $S^2 - 2cS + 1 > S^2 - 2cS + c^2 = (S - c)^2 \geq 0$.

Squaring everything gives
$$(S^2 - 2cS + 1) (3c^2 + 2c(S^2 + 1) + S)^2 < (S^2 + 2cS + 1) (3c^2 + 2c(S^2 + 1) + S)^2.$$  
Taking the difference gives
$$4c(S + 3S^2 - 9c^4S^2 + 2S^4 + 2c^2(1 - S^2 + S^4)) < 0$$

Given that $c > 0, S > 0$, we have
$$2 + 3S^2 - 9c^4S^2 + 2S^4 + 2c^2(1 - S^2 + S^4) < 0$$

Denoting $X := S^2$ and viewing the above as a quadratic polynomial of $X$, we have
$$(2c^2 + 2)X^2 + (3 - 2c^2 - 9c^4)X + (2c^2 + 2) < 0$$

Using the closed-form minimizer of quadratic polynomial (1), after some heavy algebra, one can check the minimum of (1) is
$$\frac{(1 + 3c^2)^2(1 - c^2)(7 + 9c^2)}{8(1 + c^2)}$$

which is clearly positive. Contradiction! \qed

### E.3 Proof of Lemma E.2

**Lemma E.2.** For all $0 < c < 1$ and $S > 1$,  
$$(S^2 + 3cS + 2)\sqrt{S^2 - 2cS + 1} - (S^2 - 3cS + 2)\sqrt{S^2 + 2cS + 1} > 0.$$  

**Proof.** Notice that $(S^2 + 3cS + 2) > S^2 + 2 > 0$ and $\sqrt{S^2 + 2cS + 1} > 0$. Therefore if $S^2 - 3cS + 2 \leq 0$, we are done.

Otherwise, we prove by contradiction and suppose
$$0 < (S^2 + 3cS + 2)\sqrt{S^2 - 2cS + 1} < (S^2 - 3cS + 2)\sqrt{S^2 + 2cS + 1}.$$  

26
under the condition that \( S^2 - 3cS + 2 > 0 \).

Squaring everything gives
\[
(S^2 + 3cS + 2)^2(S^2 - 2cS + 1) < (S^2 - 3cS + 2)(S^2 + 2cS + 1).
\]

Taking the difference gives
\[
cS(8 + 20S^2 - 36c^2S^2 + 8S^4) < 0
\]

Given that \( c > 0, S > 0 \), we have
\[
2 + 5S^2 - 9c^2S^2 + 2S^4 < 0
\]

Denoting \( X := S^2 \) and viewing the above as a quadratic polynomial of \( X \), we have, for \( X > 1 \),
\[
2X^2 + (5 - 9c^2)X + 2 < 0
\]

The closed-form minimizer of quadratic polynomial (2) is \( \frac{(9c^2 - 5)}{4} \). Given that \( 0 < c < 1 \), we must have \( -\frac{5}{4} < \frac{9c^2 - 5}{4} < 1 \). Hence the minimizer is not within the feasible domain \((1, \infty)\) of \( X \). Thus the minimum of (2) is achieved with \( X = 1 \) at \( 9(1 - c^2) \). This is positive. Contradiction!

\[\square\]

E.4 Proof of Fact E.3

**Fact E.3.** For a quadratic polynomial \( Ax^2 + Bx + C \) with \( A, B, C > 0 \), the minimum value on the domain \( x \geq 0 \) is \( C \), at \( x = 0 \). Therefore \( Ax^2 + Bx + C > 0 \).

**Proof.** Since \( A > 0 \), the quadratic polynomial is convex and increasing on the domain \( x > -\frac{B}{2A} \).

Since \( B > 0 \) as well, we know \( -\frac{B}{2A} < 0 \) and hence the quadratic polynomial is strictly increasing on \( x > 0 \). Therefore the minimum value is achieved when \( x = 0 \), and we obtain \( Ax^2 + Bx + C \geq C > 0 \) for all \( x \geq 0 \).

\[\square\]

**F** Examples of lazy regions

**F.1 Balanced binary classification**

We describe the data generation in Section 3.3. The label is uniformly \( \pm 1 \), that is \( P(y_i = +1) = P(y_i = -1) = 0.5 \). We have 10000 positive and negative samples \( x_i \sim \mathcal{N}(y_i, 1) \). We consider a logistic regression model \( P(Y = y|x) = \mathbb{I}(y = 1) \cdot \text{Sigmoid}(x + \theta) + \mathbb{I}(y = -1) \cdot (1 - \text{Sigmoid}(x + \theta)) = \frac{1}{1 + e^{-\gamma(x+\theta)}} \), where \( \theta \in \mathbb{R} \) is the intercept. The gradient with respect to this only trainable parameter is \( \frac{\partial c_i}{\partial \gamma} = -y \left( 1 - \frac{1}{1 + e^{-\gamma(x+\theta)}} \right) \). We set the clipping threshold \( R = 0.01 \) and the stability constant \( \gamma = 0.01 \).

**F.2 Mean estimation on Gaussian mixture data**

We also observe the lazy region issue in the mean estimation problem \( \min_{\theta} \frac{1}{2} \| \theta - x_i \|^2 \). Here \( P(x_i \sim \mathcal{N}(4, 1)) = P(x_i \sim \mathcal{N}(4, 1)) = 0.5 \). We have 10000 samples from each Gaussian distribution. The regular minimum is clearly \( \sum x_i \to 0 \), where the regular gradient and AUTO-S clipped gradient vanish. Yet both AUTO-V and Abadi’s clipping lose motivation to update the mean estimator on the interval \((-1, 1)\). We set the clipping threshold \( R = 0.01 \) and the stability constant \( \gamma = 0.1 \).

**G** Experiments settings

**G.1 Image classification settings**

We give the experiments settings for computer vision tasks in Table 1.
Figure 11: Scalar gradient $\frac{\partial L}{\partial \theta}$ at each $\theta$.

- **MNIST**: We use the network architecture from [53, 65, 61], with 40 epochs, 512 batch size, 0.5 learning rate (or 0.005 non-DP learning rate), 0.1 clipping threshold, DP-SGD with 0.9 momentum, and without pretraining. This setting is the same as [65].

- **FashionMNIST**: We use the same network architecture as MNIST, with 40 epochs, 2048 batch size, 4 learning rate (or 0.04 non-DP learning rate), DP-SGD with 0.9 momentum, and without pretraining. This setting is the same as [65].

- **CIFAR10 pretrained**: We use the SimCLR model from [13]¹⁰, with 50 epochs, 1024 batch size, 4 learning rate (or 0.04 non-DP learning rate), 0.1 clipping threshold, and DP-SGD with 0.9 momentum. The SimCLR model is pretrained on unlabelled ImageNet dataset. After pretraining, we obtain a feature of dimension 4096 on which a linear classifier is trained privately. This setting is the same as [65].

- **ImageNette**: We use the ResNet9 (2.5 million parameters) with Mish activation function [50]. We set 50 epochs, 1000 batch size, 0.0005 learning rate (or 0.000005 non-DP learning rate), 1.5 clipping threshold, and use DP-NAdam, without pretraining. This setting is the same as [35] except we did not apply the learning rate decaying scheduler.

- **CelebA (Smiling and Male and Multi-label)** We use the same ResNet9 as above, with 10 epochs, 500 batch size, 0.001 DP learning rate (or 0.00001 non-DP learning rate), 0.1 clipping threshold, and use DP-Adam, without pretraining. We use the labels ‘Smiling’ and ‘Male’ for two binary classification tasks, with cross-entropy loss. For the multi-label task uses a scalar loss by summing up the 40 binary cross-entropy losses from each label.

We refer the code for MNIST, FashionMNIST, CIFAR10, CIFAR10 pretrained to [https://github.com/ftramer/Handcrafted-DP](https://github.com/ftramer/Handcrafted-DP) by [65]. ResNet9 can be found in [https://github.com/cbenitez81/Resnet9](https://github.com/cbenitez81/Resnet9).

Throughout all experiments, we do not apply tricks such as random data augmentation (single or multiple times [16]), weight standardization [58], or parameter averaging [57].

G.2 Sentence classification settings

We experiment on five datasets in Table 2 and Table 3.

- **MNLI(m)** MNLI-matched, the matched validation and test splits from Multi-Genre Natural Language Inference Corpus.

- **MNLI(mm)** MNLI-mismatched, the matched validation and test splits from Multi-Genre Natural Language Inference Corpus.

- **QQP** The Quora Question Pairs2 dataset.

- **QNLI** The Stanford Question Answering dataset.

- **SST2** The Stanford Sentiment Treebank dataset.

¹⁰See implementation in [https://github.com/google-research/simclr](https://github.com/google-research/simclr).
The datasets are processed and loaded from Huggingface [38], as described in https://huggingface.co/datasets/glue. We follow the same setup as [74] and [40]. We refer the interested readers to Appendix G,H,I,K,N of [40] for more details.

We emphasize that our automatic clipping uses exactly the same hyperparameters as the Abadi’s clipping in [40], which is released in their Private-Transformers library.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MNLI(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch</td>
<td>18</td>
<td>18</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>Batch size</td>
<td>6000</td>
<td>6000</td>
<td>2000</td>
<td>1000</td>
</tr>
<tr>
<td>clipping threshold R</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>DP learning rate</td>
<td>5e-4</td>
<td>5e-4</td>
<td>5e-4</td>
<td>5e-4</td>
</tr>
<tr>
<td>non-DP learning rate</td>
<td>5e-5</td>
<td>5e-5</td>
<td>5e-5</td>
<td>5e-5</td>
</tr>
<tr>
<td>learning rate decay</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AdamW weight decay</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max sequence length</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
</tbody>
</table>

Table 5: Hyperparameters of automatic clipping and Abadi’s clipping, for sentence classification in Table 2 and Table 3, using either RoBERTa base or large.

Notice that we use DP learning rate 5e-4 across tasks for the R-dependent automatic DP-Adam, which is equivalent to R-independent automatic DP-Adam with the same learning rate. We demonstrate that the results are not sensitive to learning rates around the optimal choice. That is, the automatic clipping does not eliminate R at the cost of more difficult tuning of learning rate.

<table>
<thead>
<tr>
<th>learning rate</th>
<th>1e-4</th>
<th>3e-4</th>
<th>5e-4</th>
<th>8e-4</th>
<th>1e-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoBERTa-base</td>
<td>93.92</td>
<td>94.38</td>
<td>94.49</td>
<td>94.72</td>
<td>93.35</td>
</tr>
<tr>
<td>RoBERTa-large</td>
<td>95.76</td>
<td>96.21</td>
<td>96.21</td>
<td>96.33</td>
<td>95.99</td>
</tr>
</tbody>
</table>

Table 6: SST2 accuracy with respect to learning rate.

### G.3 Table-to-text generation settings

We experiment multiple GPT2 models on E2E dataset from Huggingface [38] in Table 4. We follow the same setup as [40], and our automatic clipping uses exactly the same hyperparameters as the Abadi’s clipping in [40], which is released in their Private-Transformer library.

<table>
<thead>
<tr>
<th>Model</th>
<th>GPT2</th>
<th>GPT2 medium</th>
<th>GPT2 large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Batch size</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>clipping threshold R</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>DP learning rate</td>
<td>2e-3</td>
<td>2e-3</td>
<td>2e-3</td>
</tr>
<tr>
<td>non-DP learning rate</td>
<td>2e-4</td>
<td>1e-4</td>
<td>1e-4</td>
</tr>
<tr>
<td>learning rate decay</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AdamW weight decay</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Max sequence length</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 7: Hyperparameters of automatic clipping and Abadi’s clipping, for the E2E generation task in Table 4.

---

11See https://github.com/lxuechen/private-transformers/blob/main/examples/classification/run_wrapper.py
12See https://github.com/lxuechen/private-transformers/blob/main/examples/table2text/run.sh
**H Figure zoo**

**H.1 Frequency of clipping**

We show that in all sentence classification tasks, Abadi’s clipping happens on a large proportion of per-sample gradients. This supports the similarity between Abadi’s clipping and AUTO-V in (3.1).

![Figure 12: Percentage of clipped per-sample gradients when training with DP-Adam\textsubscript{Abadi} ($\epsilon = 3$), as in Section 6.2. Left panel is RoBERTa-base and right panel is RoBERTa-large. Top row: MNLI. Middle row: QNLI. Bottom row: QQP.](image)

We note that for GPT2, GPT2 medium and GPT2 large, empirically in all iterations 100% of the per-sample gradients are clipped by the Abadi’s clipping, making the performance of Abadi’s clipping equivalent to AUTO-V clipping, as shown in Table 4.

**H.2 Stability constant helps AUTO clipping reduce gradient norm**

To corroborate our claim in Theorem 6, that the stability $\gamma$ reduces the gradient norm, we plot the actual gradient norm by iteration.

**H.3 Choice of stability constant is robust**

We claim in Theorem 6 that, as long as $\gamma > 0$ in our automatic clipping, the asymptotic convergence rate of gradient norm is the same as that by standard non-private SGD. We plot the ablation study of learning rate and the stability constant $\gamma$ to show that it is easy to set $\gamma$: in Table 2 and Table 3,
we adopt learning rate 0.0005, under which a wide range of $0.0001 < \gamma < 1$ gives similar accuracy. Note that the largest good $\gamma$ is 1000 times bigger than the smallest good $\gamma$.

Figure 14: Test accuracy by different stability constant $\gamma$ and learning rate $\eta$ in automatic clipping ($\epsilon = 3$). Upper row: SST2 for full 3 epochs. Middle row: QNLI for full 6 epochs. Lower row: QNLI for one epoch. Trained with RoBERTa-base (left) and RoBERTa-large (right).

### H.4 Automatic clipping avoids ablation study

We plot the ablation study of learning rate and clipping threshold in Abadi’s clipping below. This demonstrates that, AUTO-S clipping only requires 1D grid search to tune the learning rate, avoiding the expensive 2D grid search that is unfortunately necessary for the Abadi’s clipping. Hence our automatic clipping can save the tuning effort substantially.
We observe that GPT2 (163 million parameters), GPT2-medium (406 million), and GPT2-large (838 million), Table 4 trained with our automatic clipping consistently perform better in comparison to other methods. In some cases, LoRA trained with Abadi’s clipping also demonstrates strong performance and it would be interesting to see how LoRA trained with the automatic clipping will behave.

### J Further experiments on CelebA dataset

In this section, we present a complete summary of accuracy results, with DP constraint or not, for the CelebA dataset. We do not apply any data-preprocessing. In the first experiment, we apply a single ResNet on the 40 labels as the multi-task/multi-label learning. In the second experiment, we apply one ResNet on one label. As expected, our automatic DP optimizers have comparable test accuracy to the Abadi’s DP optimizers, but we do not need to tune the clipping threshold for each individual task/label. We also notice that, learning different labels separately gives better accuracy than learning all labels together, though at the cost of heavier computational burden.
J.1 Multi-label classification

We apply ResNet9 as in Appendix G.1 on the multi-label classification task. I.e. the output layer has 40 neurons, each corresponding to one sigmoid cross-entropy loss, that are summed to a single loss and all labels are learnt jointly.

<table>
<thead>
<tr>
<th>Index</th>
<th>Attributes</th>
<th>Abadi’s $\epsilon = 3$</th>
<th>AUTO-S $\epsilon = 3$</th>
<th>Abadi’s $\epsilon = 8$</th>
<th>AUTO-S $\epsilon = 8$</th>
<th>non-DP $\epsilon = \infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5 o Clock Shadow</td>
<td>90.64</td>
<td>90.99↑</td>
<td>90.81</td>
<td>91.28↑</td>
<td>93.33</td>
</tr>
<tr>
<td>1</td>
<td>Arched Eyebrows</td>
<td>75.15</td>
<td>76.31↑</td>
<td>76.84</td>
<td>77.11↑</td>
<td>81.52</td>
</tr>
<tr>
<td>2</td>
<td>Attractive</td>
<td>75.85</td>
<td>76.10↑</td>
<td>77.50</td>
<td>77.74↑</td>
<td>81.15</td>
</tr>
<tr>
<td>3</td>
<td>Bags Under Eyes</td>
<td>80.75</td>
<td>81.12↑</td>
<td>82.15</td>
<td>82.13↓</td>
<td>84.81</td>
</tr>
<tr>
<td>4</td>
<td>Bald</td>
<td>97.84</td>
<td>97.87↑</td>
<td>98.04</td>
<td>97.98↓</td>
<td>98.58</td>
</tr>
<tr>
<td>5</td>
<td>Bangs</td>
<td>92.71</td>
<td>92.68↓</td>
<td>93.46</td>
<td>93.55↑</td>
<td>95.50</td>
</tr>
<tr>
<td>6</td>
<td>Big Lips</td>
<td>67.51</td>
<td>67.78↑</td>
<td>68.34</td>
<td>68.44↑</td>
<td>71.33</td>
</tr>
<tr>
<td>7</td>
<td>Big Nose</td>
<td>81.92</td>
<td>80.95↓</td>
<td>83.33</td>
<td>83.28↓</td>
<td>88.55</td>
</tr>
<tr>
<td>8</td>
<td>Black Hair</td>
<td>92.25</td>
<td>92.38↑</td>
<td>93.52</td>
<td>93.09↓</td>
<td>95.49</td>
</tr>
<tr>
<td>9</td>
<td>Blond Hair</td>
<td>94.91</td>
<td>94.82↓</td>
<td>95.08</td>
<td>94.90↓</td>
<td>95.78</td>
</tr>
<tr>
<td>10</td>
<td>Blurry</td>
<td>80.13</td>
<td>82.50↑</td>
<td>83.74</td>
<td>83.89↑</td>
<td>87.79</td>
</tr>
<tr>
<td>11</td>
<td>Brown Hair</td>
<td>88.06</td>
<td>88.23↑</td>
<td>89.72</td>
<td>88.80↓</td>
<td>92.19</td>
</tr>
<tr>
<td>12</td>
<td>Bushy Eyebrows</td>
<td>94.72</td>
<td>94.54↓</td>
<td>94.54</td>
<td>94.50↓</td>
<td>95.56</td>
</tr>
<tr>
<td>13</td>
<td>Chubby</td>
<td>95.19</td>
<td>95.49↑</td>
<td>95.50</td>
<td>95.51↑</td>
<td>96.09</td>
</tr>
<tr>
<td>14</td>
<td>Double Chin</td>
<td>97.06</td>
<td>97.64↑</td>
<td>98.32</td>
<td>98.06↓</td>
<td>99.39</td>
</tr>
<tr>
<td>15</td>
<td>Eyeglasses</td>
<td>95.68</td>
<td>95.45↓</td>
<td>95.84</td>
<td>95.87↓</td>
<td>97.06</td>
</tr>
<tr>
<td>16</td>
<td>Goatee</td>
<td>96.77</td>
<td>96.79↑</td>
<td>97.02</td>
<td>97.03↑</td>
<td>98.06</td>
</tr>
<tr>
<td>17</td>
<td>Gray Hair</td>
<td>84.96</td>
<td>85.70↑</td>
<td>87.58</td>
<td>87.29↓</td>
<td>90.76</td>
</tr>
<tr>
<td>18</td>
<td>Heavy Makeup</td>
<td>81.46</td>
<td>81.42↓</td>
<td>82.62</td>
<td>82.72↑</td>
<td>86.62</td>
</tr>
<tr>
<td>19</td>
<td>High Cheekbones</td>
<td>92.05</td>
<td>92.17↑</td>
<td>93.32</td>
<td>93.17↓</td>
<td>97.46</td>
</tr>
<tr>
<td>20</td>
<td>Male</td>
<td>86.20</td>
<td>86.32↑</td>
<td>87.84</td>
<td>88.48↑</td>
<td>93.07</td>
</tr>
<tr>
<td>21</td>
<td>Mouth Slightly Open</td>
<td>96.05</td>
<td>95.96↓</td>
<td>96.08</td>
<td>95.99↑</td>
<td>96.74</td>
</tr>
<tr>
<td>22</td>
<td>Mustache</td>
<td>84.90</td>
<td>84.78↓</td>
<td>85.14</td>
<td>85.18↑</td>
<td>86.98</td>
</tr>
<tr>
<td>23</td>
<td>Narrow Eyes</td>
<td>91.55</td>
<td>91.67↑</td>
<td>92.29</td>
<td>92.45↑</td>
<td>95.18</td>
</tr>
<tr>
<td>24</td>
<td>No Beard</td>
<td>71.26</td>
<td>71.42↑</td>
<td>71.98</td>
<td>71.25↓</td>
<td>74.62</td>
</tr>
<tr>
<td>25</td>
<td>Oval Face</td>
<td>96.09</td>
<td>96.04↓</td>
<td>96.15</td>
<td>96.17↑</td>
<td>96.93</td>
</tr>
<tr>
<td>26</td>
<td>Pale Skin</td>
<td>70.34</td>
<td>72.11↑</td>
<td>72.23</td>
<td>73.01↑</td>
<td>75.68</td>
</tr>
<tr>
<td>27</td>
<td>Pointy Nose</td>
<td>93.26</td>
<td>93.02↓</td>
<td>93.56</td>
<td>93.53↓</td>
<td>94.86</td>
</tr>
<tr>
<td>28</td>
<td>Receding Hairline</td>
<td>96.16</td>
<td>96.09↓</td>
<td>96.27</td>
<td>96.46↑</td>
<td>97.44</td>
</tr>
<tr>
<td>29</td>
<td>Rosy Cheeks</td>
<td>86.39</td>
<td>87.08↑</td>
<td>88.87</td>
<td>88.63↓</td>
<td>92.25</td>
</tr>
<tr>
<td>30</td>
<td>Sideburns</td>
<td>70.30</td>
<td>71.79↑</td>
<td>73.58</td>
<td>73.19↓</td>
<td>79.15</td>
</tr>
<tr>
<td>31</td>
<td>Smiling</td>
<td>93.91</td>
<td>93.91↓</td>
<td>94.43</td>
<td>94.08↓</td>
<td>96.30</td>
</tr>
</tbody>
</table>

Table 9: Accuracy on CelebA dataset with settings in Appendix G.1 from one run. The green arrow indicates AUTO-S is better than Abadi’s clipping under the same $\epsilon$; the red arrow indicates otherwise; the black bar indicates the same accuracy.
J.2 Multiple binary classification

For the second experiment, we apply ResNet9 on each label as a binary classification task. I.e. the output layer has 1 neuron and we run 40 different models for all labels separately.

<table>
<thead>
<tr>
<th>Index</th>
<th>Attributes</th>
<th>Abadi’s Single $\epsilon = 8$</th>
<th>AUTO-S Single $\epsilon = 8$</th>
<th>Abadi’s Multi $\epsilon = 8$</th>
<th>AUTO-S Multi $\epsilon = 8$</th>
<th>non-DP Multi $\epsilon = \infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5 o Clock Shadow</td>
<td>92.15</td>
<td>92.29</td>
<td>90.81</td>
<td>91.28</td>
<td>93.33</td>
</tr>
<tr>
<td>1</td>
<td>Arched Eyebrows</td>
<td>81.18</td>
<td>80.19</td>
<td>76.84</td>
<td>77.11</td>
<td>81.52</td>
</tr>
<tr>
<td>2</td>
<td>Attractive</td>
<td>79.31</td>
<td>79.79</td>
<td>77.50</td>
<td>77.74</td>
<td>81.15</td>
</tr>
<tr>
<td>3</td>
<td>Bags Under Eyes</td>
<td>83.52</td>
<td>83.48</td>
<td>82.15</td>
<td>82.13</td>
<td>84.81</td>
</tr>
<tr>
<td>4</td>
<td>Bald</td>
<td>97.89</td>
<td>97.88</td>
<td>98.04</td>
<td>97.98</td>
<td>98.58</td>
</tr>
<tr>
<td>5</td>
<td>Bangs</td>
<td>94.52</td>
<td>94.83</td>
<td>93.46</td>
<td>93.55</td>
<td>95.50</td>
</tr>
<tr>
<td>6</td>
<td>Big Lips</td>
<td>67.32</td>
<td>67.53</td>
<td>68.34</td>
<td>68.44</td>
<td>71.33</td>
</tr>
<tr>
<td>7</td>
<td>Big Nose</td>
<td>82.31</td>
<td>82.36</td>
<td>76.69</td>
<td>80.59</td>
<td>83.54</td>
</tr>
<tr>
<td>8</td>
<td>Black Hair</td>
<td>87.08</td>
<td>86.93</td>
<td>83.33</td>
<td>83.28</td>
<td>88.55</td>
</tr>
<tr>
<td>9</td>
<td>Blond Hair</td>
<td>94.29</td>
<td>94.73</td>
<td>93.52</td>
<td>93.09</td>
<td>95.49</td>
</tr>
<tr>
<td>10</td>
<td>Blurry</td>
<td>94.95</td>
<td>95.20</td>
<td>95.08</td>
<td>94.90</td>
<td>95.78</td>
</tr>
<tr>
<td>11</td>
<td>Brown Hair</td>
<td>87.41</td>
<td>87.19</td>
<td>83.74</td>
<td>83.89</td>
<td>87.79</td>
</tr>
<tr>
<td>12</td>
<td>Bushy Eyebrows</td>
<td>91.23</td>
<td>91.43</td>
<td>89.72</td>
<td>88.80</td>
<td>92.19</td>
</tr>
<tr>
<td>13</td>
<td>Chubby</td>
<td>94.70</td>
<td>94.70</td>
<td>94.54</td>
<td>94.50</td>
<td>95.56</td>
</tr>
<tr>
<td>14</td>
<td>Double Chin</td>
<td>95.43</td>
<td>95.43</td>
<td>95.50</td>
<td>95.51</td>
<td>96.09</td>
</tr>
<tr>
<td>15</td>
<td>Eyeglasses</td>
<td>98.88</td>
<td>99.14</td>
<td>98.32</td>
<td>98.06</td>
<td>99.39</td>
</tr>
<tr>
<td>16</td>
<td>Goatee</td>
<td>96.12</td>
<td>96.07</td>
<td>95.84</td>
<td>95.87</td>
<td>97.06</td>
</tr>
<tr>
<td>17</td>
<td>Gray Hair</td>
<td>97.48</td>
<td>97.34</td>
<td>97.02</td>
<td>97.03</td>
<td>98.06</td>
</tr>
<tr>
<td>18</td>
<td>Heavy Makeup</td>
<td>88.85</td>
<td>88.72</td>
<td>87.58</td>
<td>87.29</td>
<td>90.76</td>
</tr>
<tr>
<td>19</td>
<td>High Cheekbones</td>
<td>85.66</td>
<td>85.45</td>
<td>82.62</td>
<td>82.72</td>
<td>86.62</td>
</tr>
<tr>
<td>20</td>
<td>Male</td>
<td>95.42</td>
<td>95.70</td>
<td>95.53</td>
<td>93.17</td>
<td>97.46</td>
</tr>
<tr>
<td>21</td>
<td>Mouth Slightly Open</td>
<td>92.67</td>
<td>92.74</td>
<td>87.84</td>
<td>88.48</td>
<td>93.07</td>
</tr>
<tr>
<td>22</td>
<td>Mustache</td>
<td>96.13</td>
<td>96.13</td>
<td>96.08</td>
<td>95.99</td>
<td>96.74</td>
</tr>
<tr>
<td>23</td>
<td>Narrow Eyes</td>
<td>85.13</td>
<td>85.13</td>
<td>85.14</td>
<td>85.18</td>
<td>86.98</td>
</tr>
<tr>
<td>24</td>
<td>No Beard</td>
<td>94.26</td>
<td>94.58</td>
<td>92.92</td>
<td>92.45</td>
<td>95.18</td>
</tr>
<tr>
<td>25</td>
<td>Oval Face</td>
<td>70.77</td>
<td>73.05</td>
<td>71.98</td>
<td>71.25</td>
<td>74.62</td>
</tr>
<tr>
<td>26</td>
<td>Pale Skin</td>
<td>96.38</td>
<td>96.34</td>
<td>96.15</td>
<td>96.17</td>
<td>96.93</td>
</tr>
<tr>
<td>27</td>
<td>Pointy Nose</td>
<td>71.48</td>
<td>73.37</td>
<td>72.23</td>
<td>73.01</td>
<td>75.68</td>
</tr>
<tr>
<td>28</td>
<td>Receding Hairline</td>
<td>91.51</td>
<td>91.51</td>
<td>91.75</td>
<td>91.74</td>
<td>92.87</td>
</tr>
<tr>
<td>29</td>
<td>Rosy Cheeks</td>
<td>93.26</td>
<td>93.35</td>
<td>93.56</td>
<td>93.35</td>
<td>94.86</td>
</tr>
<tr>
<td>30</td>
<td>Sidewburns</td>
<td>96.46</td>
<td>96.34</td>
<td>96.27</td>
<td>96.46</td>
<td>97.44</td>
</tr>
<tr>
<td>31</td>
<td>Smiling</td>
<td>90.82</td>
<td>90.87</td>
<td>88.87</td>
<td>88.63</td>
<td>92.25</td>
</tr>
<tr>
<td>32</td>
<td>Straight Hair</td>
<td>77.55</td>
<td>78.83</td>
<td>73.58</td>
<td>73.19</td>
<td>79.15</td>
</tr>
<tr>
<td>33</td>
<td>Wavy Hair</td>
<td>79.01</td>
<td>79.01</td>
<td>78.78</td>
<td>78.52</td>
<td>80.66</td>
</tr>
<tr>
<td>34</td>
<td>Wearing Earrings</td>
<td>87.33</td>
<td>87.50</td>
<td>82.29</td>
<td>82.20</td>
<td>87.56</td>
</tr>
<tr>
<td>35</td>
<td>Wearing Hat</td>
<td>98.04</td>
<td>98.11</td>
<td>97.46</td>
<td>97.31</td>
<td>98.68</td>
</tr>
<tr>
<td>36</td>
<td>Wearing Lipstick</td>
<td>92.05</td>
<td>90.46</td>
<td>89.87</td>
<td>90.72</td>
<td>93.49</td>
</tr>
<tr>
<td>37</td>
<td>Wearing Necklace</td>
<td>86.21</td>
<td>86.21</td>
<td>85.93</td>
<td>85.42</td>
<td>86.61</td>
</tr>
<tr>
<td>38</td>
<td>Wearing Necktie</td>
<td>95.85</td>
<td>95.94</td>
<td>94.43</td>
<td>94.08</td>
<td>96.30</td>
</tr>
<tr>
<td>39</td>
<td>Young</td>
<td>85.19</td>
<td>84.12</td>
<td>82.18</td>
<td>82.52</td>
<td>87.18</td>
</tr>
</tbody>
</table>

Table 10: Accuracy on CelebA dataset with settings in Appendix G.1 from one run. ‘Single’ means each attribute is learned separately as a binary classification task. ‘Multi’ means all attributes are learned jointly as a multi-label classification task. The green arrow indicates AUTO-S is better than Abadi’s clipping under the same $\epsilon$ and the same task; the red arrow indicates otherwise; the black bar indicates the same accuracy.
### K Code implementation of automatic clipping

Changing Abadi’s clipping to automatic clipping is easy in available codebases. One can set the clipping $R = 1$ or any other constant, as explained in Theorem 1 and Theorem 2.

#### K.1 Opacus

For Opacus [73] version 1.1.2 (latest), we can implement the all-layer automatic clipping by changing Line 399-401 in https://github.com/pytorch/opacus/blob/main/opacus/optimizers/optimizer.py to

```python
per_sample_clip_factor = self.max_grad_norm / (per_sample_norms + 0.01)
```

The per-layer automatic clipping requires changing Line 61-63 in https://github.com/pytorch/opacus/blob/main/opacus/optimizers/perlayeroptimizer.py to

```python
per_sample_clip_factor = max_grad_norm / (per_sample_norms + 0.01)
```

For older version (< 1.0, e.g. 0.15) of Opacus, we can implement the all-layer automatic clipping by changing Line 223-225 in https://github.com/pytorch/opacus/blob/v0.15.0/opacus/utils/clipping.py to

```python
per_sample_clip_factor = self.flat_value / (norms[0] + 0.01)
```

or implement the per-layer automatic clipping by changing Line 301-302 in https://github.com/pytorch/opacus/blob/main/opacus/optimizers/perlayeroptimizer.py to

```python
per_sample_clip_factor = threshold / (norm + 0.01)
clipping_factor.append(per_sample_clip_factor)
```

#### K.2 ObJAX

For ObJAX version 1.6.0 (latest), we can implement the automatic clipping in https://github.com/google/objax/blob/master/objax/privacy/dpsgd/gradient.py by changing Line 92 to

```python
idivisor = self.l2_norm_clip / (total_grad_norm + 0.01)
```

and changing Line 145 to

```python
idivisor = self.l2_norm_clip / (grad_norms + 0.01)
```

#### K.3 Private-transformers

To reproduce our experiments for sentence classification and table-to-text generation, we modify the ‘private-transformers’ codebase of [40]. The modification is in https://github.com/lxuechen/private-transformers/blob/main/private_transformers/privacy_utils/privacy_engine.py, by changing Line 349 to

```python
return self.max_grad_norm / (norm_sample + 0.01)
```

and Line 510-512 to

```python
coef_sample = self.max_grad_norm * scale / (norm_sample + 0.01)
```